

Projet Informatique Individuel

Site Vitrine

Livrable intermédiaire

27/02/2025

Sommaire

I - Description du projet.....	2
II - Les fonctionnalités requises.....	2
1) Pages requises.....	2
2) Fonctionnalités requises :	3
3) Fonctionnalités optionnelles.....	3
III - Etat des lieux.....	4
1) Maquettes.....	4
2) Implémentation de l'API.....	5
i - Choix des outils.....	5
ii - Structure et requêtes.....	6
1. Middlewares.....	6
2. Models.....	6
3. Routes.....	7
4. Serveur.....	7
5. Tests.....	8
3) Développement Front-end.....	11
i - Choix des outils.....	11
ii - Structure.....	12
1) Pages.....	12
J'ai repris mon cahier des charges pour construire les pages, je dispose donc de 4 pages :	12
2) Composants.....	12
4) Bilan, difficultés rencontrées et suite.....	13
IV - Planning.....	14
V - Annexes.....	15

I - Description du projet

J'ai choisi de construire un site internet vitrine à destination d'Emma Tremlet pour qu'elle puisse répertorier ses sites. Le site est structuré de manière à avoir une présentation d'Emma Tremlet et de son parcours, une présentation des sites déjà réalisés et une page de contact afin de la contacter pour toute question ou devis.

II - Les fonctionnalités requises

1) Pages requises

Nous avons discuté en amont avec Emma Tremlet et voici les fonctionnalités qui nous ont paru nécessaires au niveau des pages :

- Présentation
 - Introduction
 - Inclure le CV
 - Récapitulatif de ses compétences et des outils maîtrisés
- Présentation du book
 - Présentation des différents sites avec une introduction du sujet
 - Bouton ouverture du site web dans un **nouvel onglet** (certains projets ne sont pas en ligne donc pas d'ouverture juste une présentation / images et outils OU lien vers le github du projet)
- Page de contact
- Barre de navigation avec les différents onglets, et une photo ou un logo
- Footer avec :
 - lieu d'habitation, tel, email, linkedin & github
 - lien page de contact
 - plan du site
 - copyright
- Page "404 not found"

2) Fonctionnalités requises :

L'intérêt du développement de ce site est qu'il soit facilement récupérable par Mme Tremlet et qu'elle puisse rajouter facilement de nouveaux sites.

Cela implique deux choses sur le développement du site :

- La création d'une API afin de pouvoir facilement ajouter les nouveaux sites et qu'ils soient stockés.
- La création d'une interface administrateur afin de faciliter l'ajout et les modifications du site

Le site étant destiné à une développeuse, il est aussi nécessaire de simplifier au maximum le code et les explications de celui-ci, au cas où elle voudrait modifier certaines choses directement via le code.

Il est également important que le site soit responsive.

3) Fonctionnalités optionnelles

En discutant avec Emma Tremlet, nous avons également conclu qu'il serait positif de disposer de quelques fonctionnalités plus originales. L'objectif est donc d'implémenter :

- Une frise chronologique retraçant son parcours et ses formations.
L'idée est de faire apparaître ses compétences au fil de la frise
- Un autre bouton affichant des précisions sur le sujet d'un site, les outils traités, etc (en format dépliant ou en modal)
- Ajouter une page devis : faire un form sur le besoin de l'utilisateur dont le résultat doit être envoyé par mail

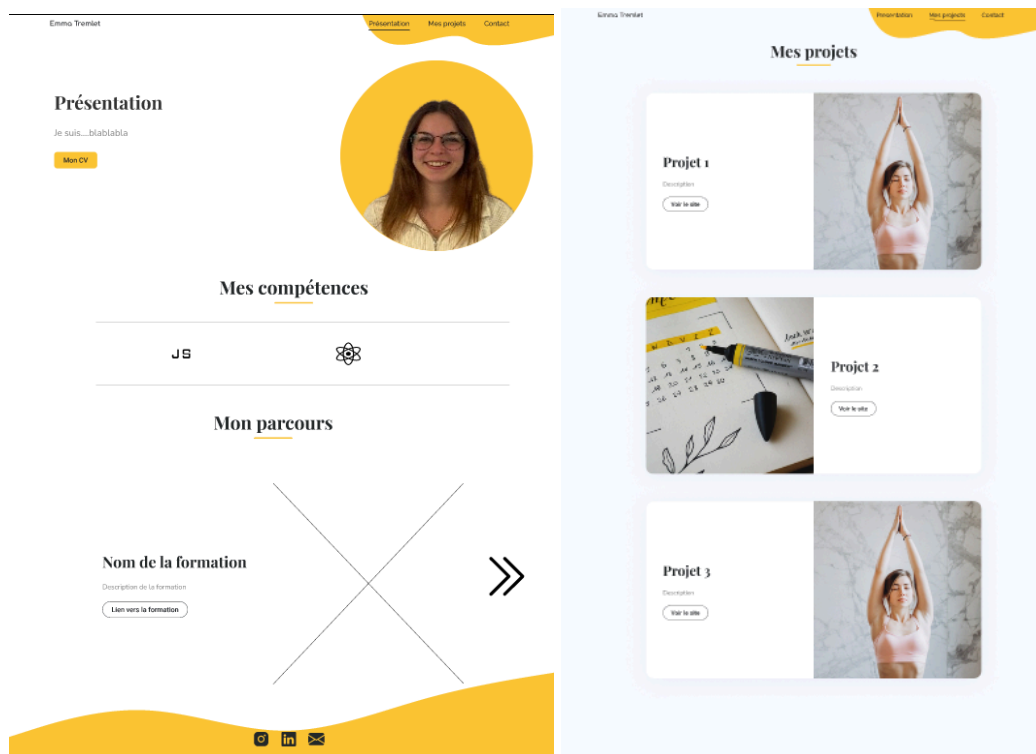
III - Etat des lieux

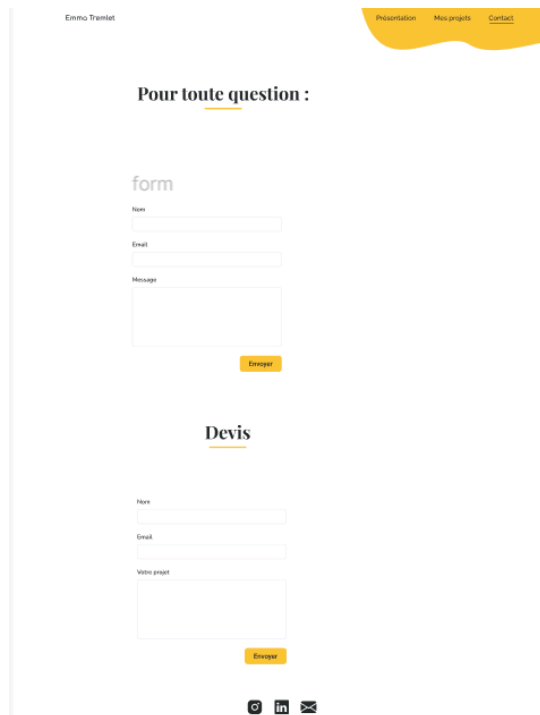
1) Maquettes

J'ai commencé par réaliser des maquettes des pages utilisateurs que je voulais obtenir afin d'avoir une vision claire de la direction du site.

J'ai donc utilisé l'outil Figma puisqu'il est très approprié au maquetage de sites internet et d'applications mobiles. Le design n'est pas gravé dans le marbre mais permet d'avoir une bonne idée du résultat attendu. Pour le style je me suis alignée avec les demandes d'Emma : un design coloré, plutôt jaune pastel, épuré.

Voici mes maquettes :





The screenshot shows a website interface for 'Emmo Tremat'. At the top, there is a navigation bar with 'Présentation', 'Mes projets', and 'Contact'. Below this, a yellow banner contains the text 'Pour toute question :'. The main content area features two forms. The first form, titled 'form', has input fields for 'Nom', 'Email', and 'Message', followed by a yellow 'Envoyer' button. The second form, titled 'Devis', has input fields for 'Nom', 'Email', and 'Votre projet', followed by a yellow 'Envoyer' button. At the bottom of the page, there are social media icons for GitHub, LinkedIn, and Email.

2) Implémentation de l'API

Étant donné que la réalisation de ce projet se déroule dans le contexte du projet informatique individuel, j'ai décidé de commencer par développer la partie API avant la partie front-end puisqu'elle représente une grosse partie de la complexité du projet.

i - Choix des outils

Pour partir sur de bonnes bases j'ai dû réfléchir à quels outils utiliser pour le code et la base de données. J'ai choisi d'utiliser les technologies MERN Stack (MongoDB, Express.js, React et Node.js). J'ai choisi de coder en React car je ne l'ai encore jamais utilisé et j'accuse donc un retard par rapport à mes camarades de promo. Ce projet est donc aussi pour moi l'occasion de me familiariser un peu plus avec cette technologie.

Cela me permet de coder en JavaScript, qui est un langage de programmation que je n'ai donc jamais pratiqué, qui est très utilisé, c'est donc une opportunité

pour moi d'en apprendre plus sur ce langage. De plus les bibliothèques React sont riches et sont très adaptées pour les sites et applications web.

J'ai privilégié MongoDB Atlas (bdd stockée sur le cloud) car dans ma base de données je n'ai pas de relations complexes, simplement des informations à stocker pour chaque site, ainsi MongoDB suffit amplement. De plus, MongoDB est déjà intégré avec Node.js et facile à mettre en place. Je n'ai utilisé jusqu'à maintenant que MySQL, j'avais donc aussi envie d'apprendre à utiliser un nouvel outil pour gérer ma base de données.

ii - Structure et requêtes

Mon API est maintenant finie et fonctionnelle, je vais présenter ici comment je l'ai structurée et les différentes requêtes que l'utilisateur, ou l'administrateur sera amené à utiliser.

1. Middlewares

J'utilise deux fichiers middlewares pour gérer la sécurité et les autorisations. J'en ai besoin car je dois différencier un utilisateur d'un administrateur, ils ne doivent pas pouvoir accéder aux mêmes commandes. J'ai donc aussi besoin d'authentifier à minima les administrateurs.

Ainsi j'ai donc :

- le fichier adminMiddleware.js qui s'occupe de vérifier si l'utilisateur est administrateur avant d'autoriser certaines actions.
- le fichier authMiddleware.js qui s'occupe de la connexion.

2. Models

J'utilise deux modèles qui sont repris dans ma base de données : un pour les utilisateurs et un pour les sites web stockés.

- User : {username, password, role} (le rôle est “user” par défaut mais peut être administrateur si on le précise dans l'ajout à la base de donnée)
- Website {name, coverImage, link, description, tools}

J'ai repris ici toutes les informations que j'ai estimé nécessaire d'avoir dans la présentation du site c'est-à-dire le nom, une image de présentation, un lien vers le site, une description du site ainsi que les outils utilisés pour sa réalisation.

3. Routes

Le dossier routes va gérer toutes les requêtes dont j'aurais besoin. J'ai séparé ce dossier en deux fichiers. Un pour gérer les requêtes liées aux sites webs de la base de données, et un pour gérer l'authentification.

J'ai donc les fichiers suivants :

- auth.js : contient les requêtes permettant de s'inscrire et de se connecter.
- website.js : contient les requêtes permettant d'ajouter, de supprimer et de modifier un site seulement si le statut d'administrateur est actif. Il permet également de consulter les sites et de les trier par identifiant (si on en cherche un en particulier) pour un utilisateur comme un administrateur.

4. Serveur

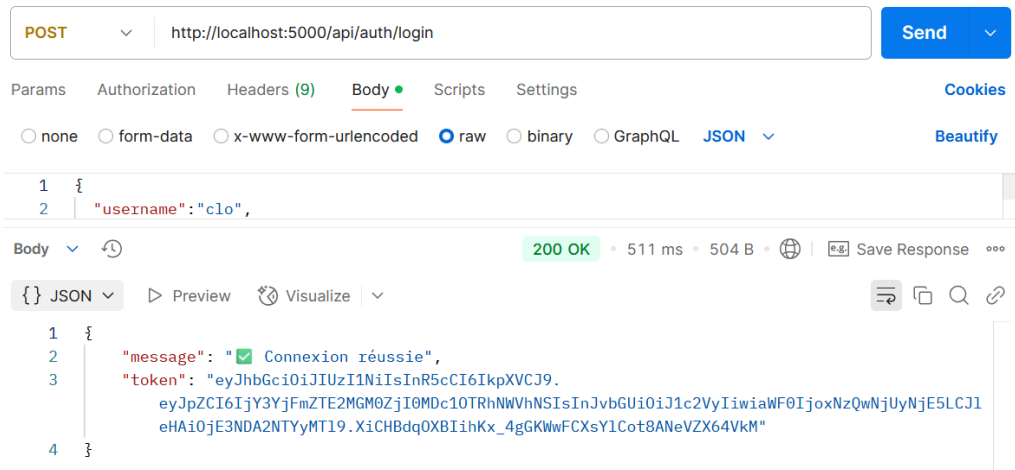
Mon serveur charge mes fichiers middlewares, models et routes et initialise la base de données. Un message de confirmation doit donc apparaître pour affirmer que la base de données est bien connectée. Ensuite il s'occupe d'utiliser les routes et de tester si le serveur fonctionne. Si c'est le cas un message s'affiche affirmant que le serveur est en ligne.

Celui-ci est ensuite démarré.

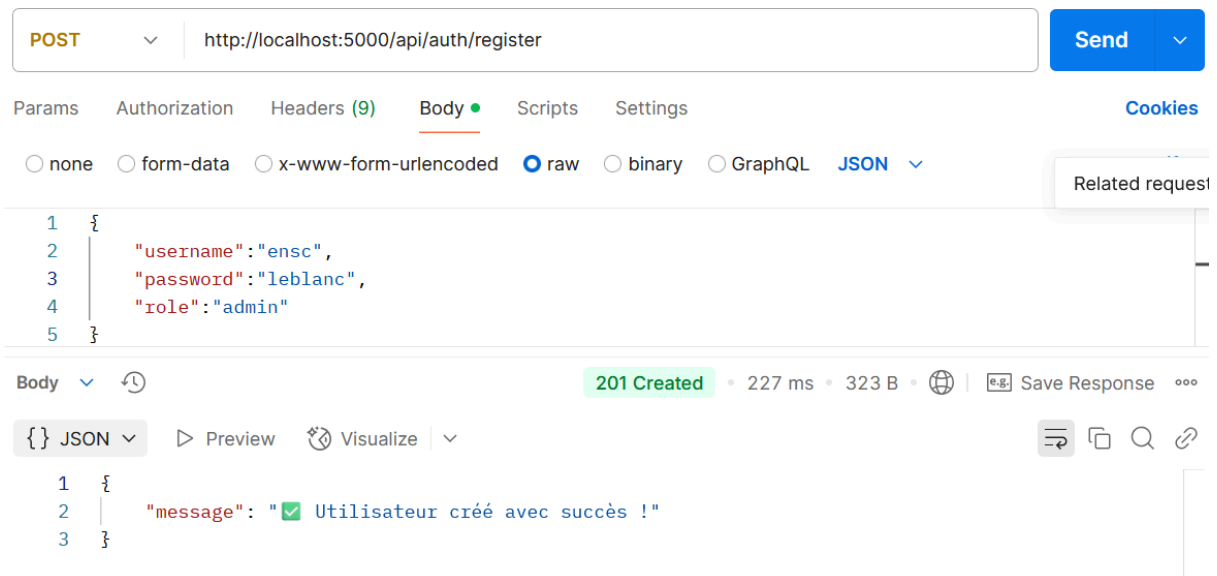
5. Tests

J'ai testé toutes mes requêtes via Postman. Voici les requêtes que j'ai testé et la preuve de leur fonctionnement :

- Login



- Register



- Register avec le statut user par défaut

POST http://localhost:5000/api/auth/register Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```

1 {
2   "username": "laura",
3   "password": "franck"
4 }

```

Body 201 Created • 214 ms • 323 B Save Response

{ } JSON Preview Visualize

```

1 {
2   "message": "✅ Utilisateur créé avec succès !"
3 }

```

- Récupération d'un site en particulier

GET http://localhost:5000/api/websites/67b1fd0e0c4f2407594a5ea0 Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Body 200 OK • 83 ms • 642 B Save Response

{ } JSON Preview Visualize

```

1 {
2   "_id": "67b1fd0e0c4f2407594a5ea0",
3   "name": "l'histoire de mona lisa",
4   "coverImage": "https://emma.com/image.png",
5   "link": "https://emma.com",
6   "description": "mona lisa est progressivement devenue lisa parce que son nom était trop
7                 long et pour un enfant c'est long et chiant de prononcer 4 syllabes pour empêcher un
8                 chat de faire des bêtises",
9   "tools": [
10    "React",
11    "Node.js"
12  ],
13   "__v": 0
14 }

```

- Récupération de tous les sites

GET http://localhost:5000/api/websites Send

Params Authorization Headers (9) Body • Scripts Settings Cookies

Body 200 OK • 68 ms • 644 B Save Response

{ } JSON Preview Visualize

```

1  [
2    {
3      "_id": "67b1fd0e0c4f2407594a5ea0",
4      "name": "l'histoire de mona lisa",
5      "coverImage": "https://emma.com/image.png",
6      "link": "https://emma.com",
7      "description": "mona lisa est progressivement devenue lisa parce que son nom était trop
                        long et pour un enfant c'est long et chiant de prononcer 4 syllabes pour empêcher
                        un chat de faire des bêtises",
8      "tools": [
9        "React",
10       "Node.js"
11     ]
12   }
13 ]

```

- Modification d'un site

PUT http://localhost:5000/api/websites/67b1fd0e0c4f2407594a5ea0 Send

Params Authorization Headers (9) Body • Scripts Settings Cookies

Body 200 OK • 77 ms • 547 B Save Response

{ } JSON Preview Visualize

```

1  {
2    "message": "Site mis à jour avec succès",
3    "website": {
4      "_id": "67b1fd0e0c4f2407594a5ea0",
5      "name": "l'histoire de Mona Lisa",
6      "coverImage": "https://emma.com/image.png",
7      "link": "https://emma.com",
8      "description": "mona lisa est un tableau super connu",
9      "tools": [
10       "React",
11       "Node.js"
12     ],
13     "v": 0
14   }
15 }

```

- Ajout d'un site

POST ▼ http://localhost:5000/api/websites/ Send ▼

Params Authorization Headers (9) **Body** • Scripts Settings Cookies

Body ▼ ↺ 201 Created • 83 ms • 539 B • 🌐 Save Response ⋮

{} JSON ▼ ▶ Preview 🔗 Visualize ▼ 🔗 🔗 🔍 🔗

```

1  {
2    "message": "Site ajouté avec succès",
3    "website": {
4      "name": "c'est notre projet",
5      "coverImage": "https://projet.com/image.png",
6      "link": "https://projet.com",
7      "description": "PARCE QUE C'EST NOTRE PROJET",
8      "tools": [
9        "React",
10       "Node.js"
11     ],
12     "_id": "67c041f03d3b7deb5be85ef5",
13     "v": 0

```

- Suppression d'un site

DELETE ▼ http://localhost:5000/api/websites/67c0422d3d3b7deb5be85ef7 Send ▼

Params Authorization • Headers (10) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

Body ▼ ↺ 200 OK • 61 ms • 308 B • 🌐 Save Response ⋮

{} JSON ▼ ▶ Preview 🔗 Visualize ▼ 🔗 🔗 🔍 🔗

```

1  {
2    "message": "Site supprimé avec succès"
3  }

```

3) Développement Front-end

Le développement du front-end est commencé mais est loin d'être terminé. Pour le moment la structure est prête mais il reste encore à relier l'API pour chercher les documents nécessaires dans la base de données et les afficher via le site. Le style du site est entamé mais reste à ses débuts. Cela constitue une priorité pour la suite du projet.

i - Choix des outils

Pour le développement front-end j'ai décidé de coder avec React car à terme j'aimerais intégrer des animations (frise chronologique, formulaires, boutons

animés). De plus, cela donne une facilité pour le faire évoluer si par exemple Emma Tremlet voudrait ajouter par la suite une boutique ou un espace commentaires, l'espace est déjà prêt. Cela permet d'avoir également une bonne structure du code avec des éléments et des pages bien définis. Avoir un site dynamique est un objectif de ce projet, ainsi React semble être une bonne solution.

J'ai choisi d'utiliser la bibliothèque Tailwindcss. Tout d'abord cette librairie est compatible avec React et elle permet d'avoir un style css cohérent et créatif. Puisqu'un site vitrine doit refléter aussi la personnalité du vendeur, il est important de pouvoir avoir une identité visuelle originale et cette librairie le permet de manière simple et efficace.

ii - Structure

1) Pages

J'ai repris mon cahier des charges pour construire les pages, je dispose donc de 4 pages :

- Home.jsx qui correspond à la page d'accueil
- Projects.jsx qui correspond à la page qui répertorie les sites webs
- Contact.jsx qui correspond à la page de contact
- 404.jsx qui correspond à la page d'erreur 404

2) Composants

J'utilise plusieurs composants qui sont repris dans plusieurs pages différentes :

- Navbar.jsx qui correspond à la barre de navigation qui est utilisée sur toutes les pages
- Footer.jsx qui correspond au footer qui est utilisé pour toutes les pages
- ProjectCard.jsx qui correspond à la présentation d'un site web qui est repris sur la page Projects.jsx
- ContactForm.jsx qui correspond au formulaire de contact utilisé sur la page Contact.jsx

Pour le composant ProjectCard j'aimerais utiliser la méthode `fetch()` pour appeler l'API depuis le fichier `.jsx` mais cette fonctionnalité n'est pas encore prête.

4) Bilan, difficultés rencontrées et suite

Pour conclure, l'API est opérationnelle et le site est en cours de construction. Les parties dynamiques ne sont pas encore commencées et il faut que je relie les documents de ma base de données à mon site dans ma page présentant les projets.

J'ai rencontré plusieurs difficultés au cours de ce début de projet, qui se sont surtout concentrées sur les outils.

- J'ai eu des difficultés à prendre en main et à comprendre comment utiliser Node, Express et MongoDB. Cela m'a valu de perdre du temps surtout concernant les tests à cause de MongoDB. En effet le code n'avait rien de problématique mais les requêtes ne voulaient pas fonctionner. Il s'agissait d'un problème de configuration de MongoDB pour rendre accessible la base de données pour n'importe quelle adresse IP.
- J'ai également rencontré des soucis lors de l'installation de la librairie Tailwindcss. Je n'arrivais pas à l'installer et je ne comprenais pas les erreurs malgré la consultation de la documentation disponible. Je pensais qu'il s'agissait d'un conflit entre deux versions (une installée par défaut sur mon espace de développement et une que j'essayais d'installer). J'ai tenté de résoudre les conflits, cela n'a pas fonctionné. Grâce aux recherches de mon tuteur j'ai pu comprendre que cela venait des propriétés du projet Create React App qui est incompatible avec certaines configurations. Grâce à la documentation de TailwindCSS nous avons pu trouver une solution. Cela a considérablement retardé mon avancée sur le développement du site, puisque la partie css n'avancait pas.
- J'ai également mis du temps pour prendre en main les outils. En effet, n'ayant pas codé en web depuis plus d'un an et n'ayant pas de bases sur les technologies MERN Stack, j'ai mis du temps à retrouver mes réflexes et à me familiariser avec ces outils. Sous conseil de mon tuteur, j'ai entamé des formations gratuites sur OpenClassroom pour renforcer les bases sur React, MongoDB et Node. Cependant, les formations étant conséquentes, ayant un rythme de cours plutôt intense et voulant avancer sur le projet, je n'ai pas beaucoup avancé sur ces formations.

IV - Planning

Après notre première réunion avec mon tuteur Mr François DOMEQ, j'ai révisé mon planning prévisionnel initial en tenant compte de ses conseils.

Les cases surlignées en vert sont les fonctionnalités faites, celles en orange sont celles en cours et enfin celles qui sont blanches sont les fonctionnalités pas encore commencées.

Planning prévisionnel PII													
Planning prévisionnel PII													
Etapas	Détails	JANVIER			FEBVIER			MARS			AVRIL		
		15-19 janvier	20-27 janvier	27 janvier-3 février	3-10 février	10-17 février	17-24 février	24 février-3 mars	3-10 mars	10-17 mars	17-24 mars	24-31 mars	1-7 avril
Front-end static	Préparation des outils de développement												
	Page accueil												
	Page book + utilisation des données de l'API												
	Développement de la barre de navigation et du footer												
	Page devis (formulaire statique sans logique)												
Front end dynamic	Page "404 not found"												
	Frise chronologique												
	Modèles ou dépliant												
	Boutons et liens												
	Responsive design												
Back-end	Préparer les outils de développement												
	Implémentation BDD												
	Gestion utilisateurs												
	Fonctionnalités Create, Read, Update, Delete												
	Tests												
Connexion front & Back	Intégrer la gestion des utilisateurs (connexion, accès administrateur)												
Maquettes	Maquette accueil												
	Maquette book												
	Maquette devis												
	Maquette page de contact												
Préparation Soutenance	Front end static												
Résolution bugs	Back end												
	Connexion à l'API												
	Front end dynamic												
Formation	Formation à React												
	Formation à MongoDB												
	Formation Node												

Par rapport à mon planning initial j'y ai donc rajouté un temps de formation, rallongé le temps de développement du site et rajouté un temps de préparation pour rendre ce livrable.

Cependant j'ai rencontré des difficultés (présentées plus haut) et voici le planning réel jusqu'à ce jour et les prévisions pour la suite compte tenu de mon avancement:

Planning prévisionnel PII													
Planning prévisionnel PII													
Etapas	Détails	JANVIER			FEBVIER			MARS			AVRIL		
		15-19 janvier	20-27 janvier	27 janvier-3 février	3-10 février	10-17 février	17-24 février	24 février-3 mars	3-10 mars	10-17 mars	17-24 mars	24-31 mars	1-7 avril
Front-end static	Préparation des outils de développement												
	Page accueil												
	Page book + utilisation des données de l'API												
	Développement de la barre de navigation et du footer												
	Page devis (formulaire statique sans logique)												
Front end dynamic	Page "404 not found"												
	Frise chronologique												
	Modèles ou dépliant												
	Boutons et liens												
	Responsive design												
Back-end	Préparer les outils de développement												
	Implémentation BDD												
	Gestion utilisateurs												
	Fonctionnalités Create, Read, Update, Delete												
	Tests												
Connexion front & Back	Intégrer la gestion des utilisateurs (connexion, accès administrateur)												
Maquettes	Maquette accueil												
	Maquette book												
	Maquette devis												
	Maquette page de contact												
Préparation Soutenance	Front end static												
Résolution bugs	Back end												
	Connexion à l'API												
	Front end dynamic												
Formation	Formation à React												
	Formation à MongoDB												
	Formation Node												

J'ai pris plus de temps que prévu sur le développement de l'API à cause des bugs sur la période de tests. J'ai également été retardée pour le développement de la

partie static du site à cause des bugs d'installation de la librairie que je souhaite utiliser.

Je souhaite donc pouvoir finir le développement static du site d'ici la semaine prochaine et pouvoir commencer à intégrer des éléments dynamiques. Je souhaite également avancer sur les formations au besoin.

V - Annexes

Je vous joint en annexe mon repository github pour accéder à mon code source :

<https://github.com/louisetaupin/PII>

Je vous joint aussi mon planning si vous souhaitez plus de détails :

https://docs.google.com/spreadsheets/d/1TFMxFWAammrqvHtEl4ELoqzJb8FZzGoNVow5vua_5CY/edit?usp=drive_link