|  |
| --- |
| IUT DIJON AUXERE |
| Rapport Projet tutoré S4 |
| Santech |

|  |
| --- |
|  |



Année scolaire 2020-2021

Iris Boclet-Richter   
Mathys Clerget  
Antonia De Oliveira  
Manon Fromain -- Gardaz  
Albane Maublanc  
Thibault Trembleau

Table des matières

[I. INTRODUCTION 2](#_Toc67410319)

[II. LE PROJET 2](#_Toc67410320)

[A. Conception 2](#_Toc67410321)

[Objectifs 2](#_Toc67410322)

[Maquettes 3](#_Toc67410323)

[Choix des technologies 3](#_Toc67410324)

[B. Réalisation 4](#_Toc67410325)

[Répartition 4](#_Toc67410326)

[Difficultés rencontrées 4](#_Toc67410327)

[Utilisations des technologies choisies 4](#_Toc67410328)

[C. Produit Final 5](#_Toc67410329)

[Réalisation des objectifs 5](#_Toc67410330)

[Apports du projet 5](#_Toc67410331)

[Tutoriel d’utilisation 6](#_Toc67410332)

[III. CONCLUSION 7](#_Toc67410333)

Table des illustrations

[Image 1 Maquettes 3](file:///D:\Document\cours\IUT_INFO\S4\Ptut\Git\Rapport%20ptut.docx#_Toc67062588)

# INTRODUCTION

Ce projet tutoré nous a été inspiré du contexte sanitaire actuel. Nous avons voulu réaliser un carnet de santé numérique qui permettrait à l’utilisateur de suivre son profil santé en permanence en sachant quels sont ses vaccins récents ou son temps moyen de sommeil par exemple.

Nous avons décidé de présenter ce carnet sous la forme d’un site internet pour permettre l’accessibilité depuis un portable, un ordinateur ou encore une tablette selon les facilités de la personne l’utilisant. (Nous avons, notamment lors de ce choix, envisagé la possibilité de l’usage de notre projet par des personnes âgés qui ne sont pas forcément à l’aise avec un téléphone et à qui un site conviendrait donc mieux qu’une application mobile.)

Les droits de récupération et utilisation de données sur la santé en France étant assez compliqués ce projet ne sera pas disponible au public pour le moment car il demanderait un travail juridique plus poussé.

Ce projet est cependant fonctionnel si la partie juridique est mise de côté.

Nous avons choisi de réaliser ce projet avec les technologies de React[[1]](#footnote-1) pour la partie front end et de Node1 et MongoDB [[2]](#footnote-2) car ce sont des technologies que nous ne connaissions pas et que nous voulions donc étudiez de notre côté. De plus ce sont des technologies très utilisées pour les sites internet et elles nous paraissaient donc adaptées à notre besoin.

# LE PROJET

## Conception

### Objectifs

Nos objectifs avec ce projet étaient de fournir un carnet de santé numérique complet pour permettre aux utilisateurs d’avoir accès à ces informations n’importe où et n’importe quand. Apprendre les technologies mentionné en introduction était aussi un des grands objectifs de ce projet.

Nous avions donc comme projet d’inclure dans ce site :

* Des graphiques concernant le temps de sommeil, le poids et la taille, le cycle des règles (pour les personnes concernées) et enfin les dons (sang, plasma, plaquettes) effectués.
* Des objectifs que l’utilisateur pourrait se fixer (ex: « perdre 2 kg »).
* Un calendrier qui permettrait un aperçu du cycle des règles et de la date de fin prévu des objectifs. Nous avons aussi pensé à inclure la possibilité de marquer des événements (ex : « rendez-vous chez le médecin à 14h »).
* Des astuces de santé (génériques, non ciblés en fonction du profil de chaque utilisateur) sur la page d’accueil.
* Un profil qui permettrait de voir l’âge, le sexe, les vaccins effectués et les allergies de l’utilisateur.
* Comme nous voulions rendre le site un peu ludique pour inciter les utilisateurs à prendre soin de leur santé nous voulions aussi ajouter un système de succès (ex : « avoir eu 3 jours de suite avec un temp de sommeil supérieur à 8h ») ainsi qu’un système d’expérience (« débloquer des succès, faire des dons, etc. augmenteraient l’expérience ») et de niveau.

### Maquettes

Après avoir défini ces objectifs nous avons défini des maquettes (sans pour le moment se préoccuper des couleurs) des principaux composants de notre site.

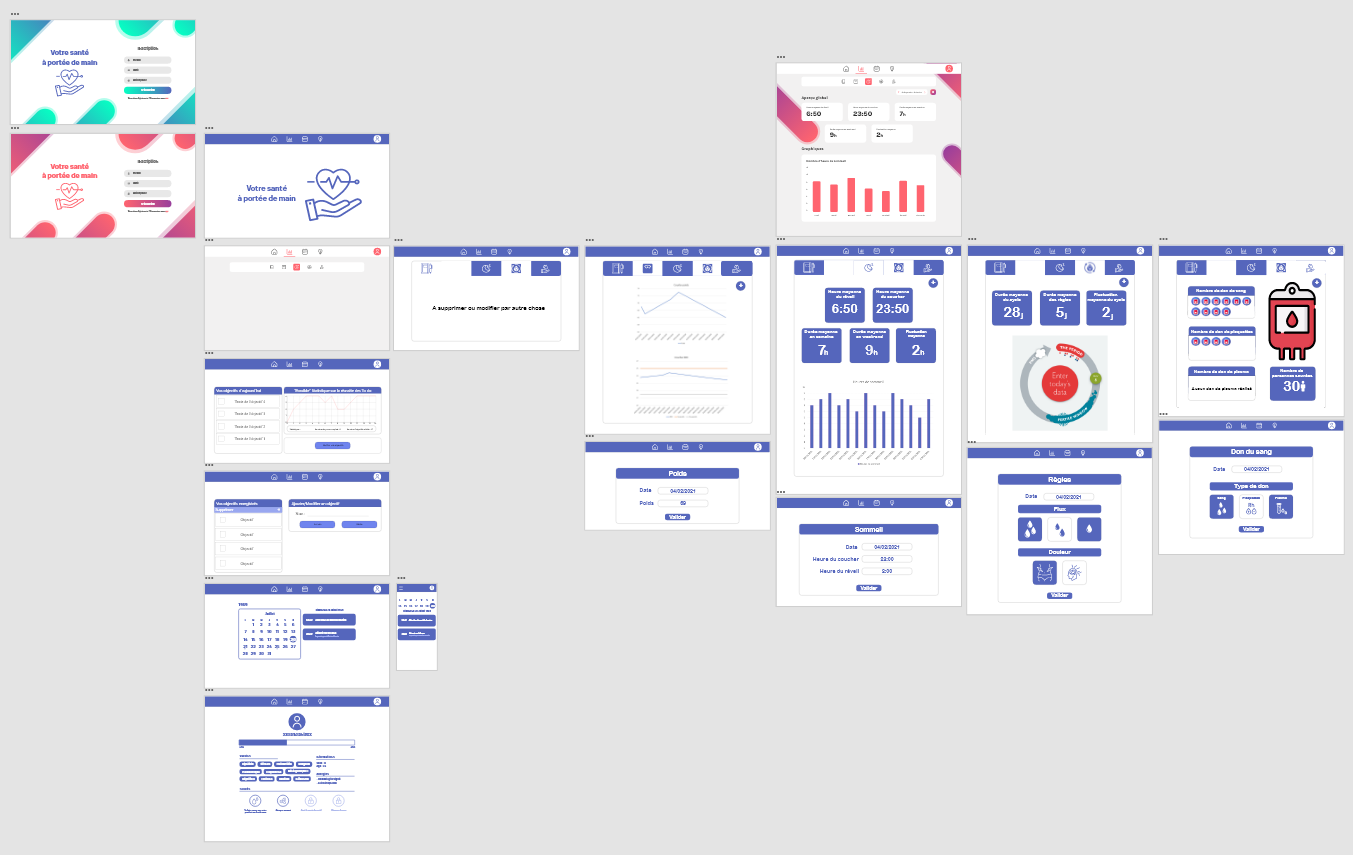


Image Maquettes

Nous avons après réalisé une charte graphique que voici :

### Choix des technologies

Nous avons décidé d’étudier des technologies Web pour créer ce projet. De toutes nouvelles technologies qui nous étaient encore inconnues jusqu’ici.

Tout d’abord pour la partie frontend nous avons recouru à l’utilisation d’une bibliothèque JavaScript qui s’appelle React, qui est développé et utilisé par Facebook. En effet, les applications web utilise maintenant ce genre de bibliothèque pour rendre les sites plus dynamiques. React va nous aider dans la construction d’un projet structuré dans la partie frontend. De plus il nous permet de faire une « single page application » c’est-à-dire que notre site n’aura pas besoin de se recharger pour changer de composants (par exemple pour changer de page).

En parallèle de la partie frontend nous avons utilisé dans la partie backend la plateforme Node.js qui nous permet de faire du JavaScript côté serveur. Avant nous avions l’habitude de faire du PHP pour la partie serveur, Node.js nous permet d’être plus flexible sur notre code. Par ailleurs, avec Node.js (couplé avec la bibliothèque Express.js qui nous permet de facilité la programmation) nous avons la possibilité de faire du FullStack (faire du frontend[[3]](#footnote-3) et backend[[4]](#footnote-4)) javascript comme nous le couplons avec React. De plus nous avons accès à un ensemble d’outils mise à disposition par la communauté facilitant la programmation.

Finalement, pour ce qui est du langage pour la base de données, nous avons pris MongoDB au lieu du SQL que nous avions étudié jusqu’à présent. En plus de nous apprendre une nouvelle technologie, MongoDB s’associe bien avec des projets javascript de part de son utilisation de JSON (Format de données textuelle).

## Réalisation

### Répartition

Pour avancer efficacement dans ce projet nous avons organisé une réunion hebdomadaire dans laquelle nous définissions des tâches à faire pour la semaine suivante. Après les tâches définis chacune d’entre elles étaient attribuées à une jusqu’à trois personnes et chaque personne avait une à trois tâches.

Nous avons essayé de faire une répartition de sortes à ce que tout le monde touche au backend et au frontend pour pouvoir s’approprier les nouvelles technologies que nous utilisions.

*Diagramme de Gant*

### Difficultés rencontrées

Pendant ce projet nous avons au départ rencontrés quelques difficultés liées aux technologies que nous utilisions qui n’étaient pas forcément intuitives mais nous avons assez vite réussi à nous les approprier dans l’ensemble.

En plus de cela, pour l’authentification d’un utilisateur nous avions pour habitude d’utiliser des sessions avec PHP. Ici nous avons eu recours à des tokens d’authentification (standard JSON Web Token). C’était un sujet assez complexe que nous avons mis du temps à prendre en main pour rendre l’authentification la plus sécurisé possible. Avec un peu de temps et de patience nous avions enfin pu résoudre ce problème.

Nous n’avons pas rencontré d’autre difficultés majeures.

### Utilisations des technologies choisies

Ici vous trouverez les technologies ainsi que les outils extérieurs que nous avons utilisé dans notre projet et leurs utilités :

* [React](https://fr.reactjs.org) : Bibliothèque pour la partie Front End/Client
  + [React Router](https://reactrouter.com) : Bibliothèque de routage des URL (Permet de charger des composants en fonction de l'URL)
  + [Axios](https://www.npmjs.com/package/axios) : Permet de faire des requêtes au serveur plus simplement (GET, POST...)
  + [Chart.js](https://www.chartjs.org) : Bibliothèque pour créer des graphiques
* [Node.js](https://nodejs.org/fr/) : Environnement d'exécution qui permet d'utiliser du JavaScript côté serveur
  + [npm](https://www.npmjs.com) : Gestionnaire de paquets de Node.js (installé automatiquement avec Node.js)
  + [Express.js](https://expressjs.com/fr/) : Framework qui fournit des outils pour faciliter le développement d'applications Node.js
  + [Mongoose](https://mongoosejs.com) : Bibliothèque pour des applications Node.js qui utilisent MongoDB
  + [JWT](https://jwt.io) : JSON Web Token utilise des tokens pour échanger des informations entre plusieurs tiers. Par exemple pour s'authentifier.
  + [CORS](https://www.npmjs.com/package/cors) : Permet de pouvoir partager des ressources entre deux serveurs (par exemple entre le client et le serveur)
  + **(petit paquet)** [Cookie-parser](https://www.npmjs.com/package/cookie-parser) : Permet de lire les cookies de notre application
  + **(petit paquet)** [Multer](https://www.npmjs.com/package/multer) : Un middleware pour upload des images
  + **(petit paquet)** [Nodemon](https://nodemon.io) : Utilitaire qui permet de redémarrer l'application Node lors d'un changement dans le code
  + **(petit paquet)** [Dotenv](https://www.npmjs.com/package/dotenv) : Module qui permet de charger des variables d'environnements de fichier **.env** en **process.env.** dans le code
  + **(petit paquet)** [Validator](https://www.npmjs.com/package/validator) : Bibliothèque pour valider des string (par exemple des mails)
  + **(petit paquet)** [Bcrypt](https://www.npmjs.com/package/bcrypt) : Bibliothèque pour le hash des mots de passe
* [MongoDB](https://www.mongodb.com/fr) : Base de données NoSQL orientée documents (pas de table comme MySQL)

## Produit Final

### Réalisation des objectifs

Par rapport aux objectifs que nous nous étions fixés au départ nous avons réussis à réaliser :

Nous avons malheureusement dû laisser de côtés :

La non-réalisation de ces fonctionnalités ont été causée par un manque de temps et nous avons dû jugés que nous préfèrerions laisser des fonctionnalités moins importantes de côtés et rendre un projet moins complet mais fonctionnel.

### Apports du projet

Ce projet nous a permis d’apprendre à gérer la réalisation d’un projet en complète autonomie ce qui était la première fois pour la plupart d’entre nous. Comme nous n’avions pas cette fois de professeur référant nous avons développé notre capacité à réaliser un projet de nous-même de sa conception à sa finalisation.

Nous avons donc renforcé nos capacités d’organisation grâce à ce projet.

D’un point de vue plus technique nous avons aussi appris l’utilisation de 3 nouvelles technologies : React.js, Node.js et MongoDB comme évoqués plusieurs fois dans ce rapport.

Le temps passé sur ce projet ne nous a pas permis de les maitriser sur le bout des doigts mais nous avons maintenant une connaissance générale de cette bibliothèque pour le frontend, de cette plateforme pour gérer le serveur et de ce système de gestion de base de données. Cela nous permet ainsi d’avoir une connaissance suffisante pour développer un site web possédant une partie serveur uniquement en JavaScript. Connaissance que nous n’avions pas avant ne sachant faire que des sites en PHP.

Enfin, ce projet nous a bien sur permis de développer grandement notre connaissance du javascript que nous avons commencés à étudier en cours à peu près en même temps que le démarrage de ce projet.

### Tutoriel d’utilisation

Notre site n’est pour le moment pas disponible au public mais est accessible seulement en passant par des serveurs locaux qui nécessite d’avoir installé **Node.js**.

Avant toutes choses vous devez donc installer **Node.js** qui est un environnement pour exécuter du Javascript. Pour cela, téléchargez la dernière version sur <https://nodejs.org/fr/>. Vérifier que Node.js est bien installé, vous pouvez ouvrir l’invite de commande (Touche Windows + recherche « cmd » sur Windows) et tapez « node -v », vous devez voir afficher la version de node que vous avez installé.

Ensuite, enregistrez le chemin où se trouve les dossiers de notre projet et dans l’invite de commande tapez « cd C:/votreChemin/ptuts4\_santech/Application/ » (à modifier selon votre chemin).

Pour lancer le serveur (ou backend), vous allez devoir accéder au dossier server en faisant « cd server » dans l’invite de commande, ensuite, toujours dans l’invite de commande, faites « npm start » pour lancer le serveur.

Maintenant, ouvrez une autre invite de commande (en laissant la précédente ouverte) et accédez au dossier client dans le dossier Application avec la commande « cd chemin/client» comme précédemment. Ensuite lancez le côté client en tapant « npm start », vous devriez voir une page de votre navigateur s’ouvrir avec notre projet d’affiché (cela peut prendre un peu de temps).

Voilà, notre projet est à votre disposition (ne fermez pas les deux invites de commande quand vous testez le site). Vous devrez recommencez les étapes si vous avez fermez les invites de commande.

# CONCLUSION

En conclusion ce projet tutorés de S4 nous aura permis de développer une maturité quant à la conception et réalisation d’un projet en autonomie ainsi qu’une connaissance sur de nouvelles technologies.

Nous n’avons malheureusement pas eu le temps d’implémenter toutes les fonctionnalités que nous voulions y intégrer au départ mais c’était un sacrifice à faire pour rendre un livrable fonctionnel.

WE FAUDRA METTRE PLUS DE TRUC

1. Bibliothèque Javascript [↑](#footnote-ref-1)
2. Système de gestion de base de données NoSQL [↑](#footnote-ref-2)
3. Partie client d’un site web, c’est-à-dire la partie qui apparaitra sur le navigateur de l’utilisateur. [↑](#footnote-ref-3)
4. Partie serveur d’un site web (liens avec les bases de données etc.). [↑](#footnote-ref-4)