

Rapport Projet Tuteuré

Projet MiMi :

Comment simuler la présence d'un élève en classe ?

Lucas Abitbol
Wassim Ait Youcef
Léa Fournier
Baptiste Villedieu
FIE4 – Promotion 2023

Sommaire

Introduction.....	3
Contexte et définition du sujet.....	4
Gestion de projet	5
Développement de l'application	6
1) Stratégie de développement.....	6
2) Application côté client.....	7
3) Application côté serveur.....	8
Description de la solution	11
Fonctionnalités restantes à développer	13
1) La visioconférence.....	13
2) Relier le front end et le back end	13
3) Développement d'un jeu multijoueur	13
4) Choix de l'avatar	14
5) Stockage des fichiers	14
Conclusion	15
Bibliographie	16
Annexes.....	17

Introduction

L'école est un véritable lieu d'éducation, de socialisation et de vie. À l'école, on découvre une grande partie des aspects importants de la vie. Les premières rencontres, les premières amitiés, les premières disputes. L'école est plus qu'un lieu où l'on apprend à lire et à écrire, c'est un lieu où l'on apprend à vivre, à découvrir les autres et à se découvrir soi-même.

Mais malheureusement, quand certains s'épanouissent dans le milieu scolaire et profitent de leurs jeunes années, d'autres ne se sentent pas à leur place. D'autres n'ont pas cette chance de pouvoir s'épanouir à l'école. Handicaps, maladies, phobies scolaires, tout autant de raisons qui vont pousser un enfant à ne plus se rendre à l'école pendant de longues durées.

Des solutions ont été mises en place pour aider ces enfants. En effet, l'Éducation Nationale a mis en place des aides, des professeurs à domicile ou à l'hôpital afin de dispenser les cours. Mais le nombre de places est limité et une grosse problématique n'est pas résolue. Car si, via cette solution, les enfants bénéficient de l'aspect pédagogique, purement académique de l'école, ils ne bénéficient pas de l'aspect le plus important, l'aspect social. Ce dernier, trop souvent négligé, est souvent celui qui pousse beaucoup d'enfants et d'adolescents à se lever le matin. Si ce dernier est négligé, la scolarité de l'enfant risque d'en pâtir énormément. Les enfants se retrouvent livrés à eux même, sans aucun soutien.

Pour améliorer cela, il existe des PEP (Pupilles de l'Enseignement Public). Il s'agit d'un réseau d'associations locales qui s'occupent des décrocheurs précoces non pris en charge par les solutions gouvernementales. Au niveau départemental, l'ADPEP (Association Départementale des Pupilles de l'Enseignement Public) s'appuie sur un service d'accompagnement scolaire dénommé SAPAD (Service d'Assistance Pédagogique À Domicile).

Cependant, le mécanisme d'action de SAPAD est le même que celui d'APADHE (Accompagnement Pédagogique À Domicile, à l'Hôpital ou à l'École). Dans ce cas, la question du maintien des liens sociaux avec tous les camarades de classe n'est pas respectée. Cela va à l'encontre des objectifs fondamentaux de ce service. Même si des expériences sont faites pour pallier ce problème, elles ne sont pas définitives. Par exemple, une solution utilisant un robot de téléprésence a été testée, mais elle a trop distrait les élèves de la classe.

C'est alors que l'AD-PEP du Tarn a contacté le CHL afin d'arriver à trouver un accord concernant la conception et le développement d'une solution numérique capable de permettre à l'élève d'interagir avec sa classe et ses professeurs tout en suivant les cours à distance.

La solution et les précédents travaux réalisés ont été expliqués dans un précédent rapport. Ce rapport va donc décrire et détailler nos travaux réalisés dans le cadre du projet tuteuré de 4A menés par ABITBOL Lucas, AIT YOUCEF Wassim, FOURNIER Léa et VILLEDIEU Baptiste.

Contexte et définition du sujet

Lors du premier semestre s'étant déroulé de septembre à décembre 2021, nous avons pour mission de conceptualiser le projet initial. En effet, le projet avait été déjà entamé l'année dernière dans le cadre d'un autre projet tuteuré, ainsi que d'un stage de fin d'étude. Nous avons repris les travaux déjà effectués afin de les analyser et d'en tirer des conclusions pertinentes dans le but de préparer au mieux le développement de la solution.

Nous nous sommes notamment basés sur l'enquête de terrain qui a été réalisée en amont. Cette enquête visait à recueillir l'avis de différents acteurs vis-à-vis de la solution envisagée. Que ce soit un robot, une tablette ou tout autre moyen, cette enquête nous a permis de dégager les retours de la population concernée.

À la suite de ces entretiens, nous en avons conclu que l'outil doit être le plus discret possible afin qu'il s'intègre au mieux dans la classe. Les étudiants ne doivent pas se désintéresser du cours pour la machine ou la solution en place. Il doit aussi pouvoir être déplaçable facilement quel que soit l'environnement de classe (changement de salle de cours, cours de sport en extérieur). Il faut également assurer la continuité du cours au mieux via la solution tout en l'agrémentant de fonctionnalités qui permettent de maintenir le lien social. Des jeux multi-joueurs ou encore le "chuchotement" lors des visioconférences doit pouvoir être possible.

Après étude et analyse des éléments à notre disposition, nous en sommes venus à choisir la solution qui, selon notre analyse, remplit au mieux les conditions fixées.

Gestion de projet

Au premier semestre, nous avons adopté une méthode de gestion de projet Agile, où nous faisons régulièrement des réunions pour planifier ce qu'il faudrait faire dans les prochains jours.

Ce semestre, nous avons décidé d'adopter une méthode différente. En effet, nous avons compris que nous devrions nous séparer le travail pour améliorer le développement de l'application.

Après discussion, Léa et Wassim se sont dirigés sur le développement du front end. En effet, ils apprécient produire des visuels et ce côté du projet les attirait plus. Au contraire, Lucas et Baptiste ont préféré s'orienter vers le développement du back end.

Nous avons également remarqué que les objectifs visés au premier semestre seraient difficiles à atteindre.

Beaucoup d'heures de PTUT n'ont pas pu être faites au début du semestre et le fait de devoir apprendre deux nouveaux langages de programmation nous a fait perdre beaucoup de temps.

Nous avons donc pu générer le diagramme de Gantt du 2^{ème} semestre ci-dessous.

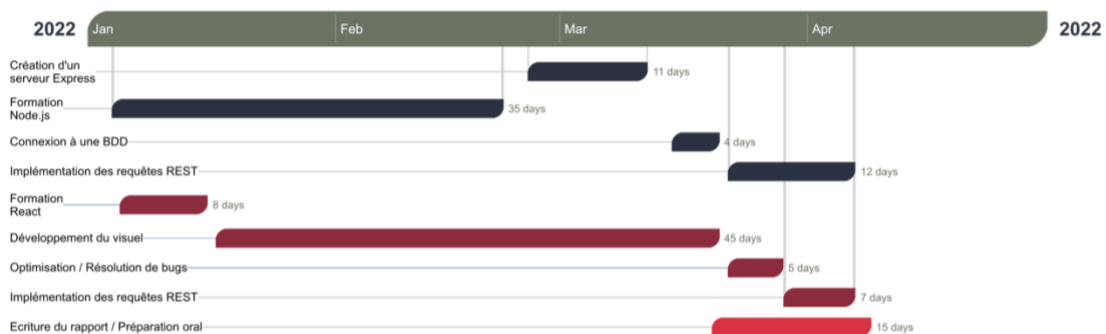


DIAGRAMME DE GANTT SEMESTRE 2

Développement de l'application

Il faut savoir qu'à l'origine, la solution choisie avait pour objectif principal d'être simple. Elle devait être non seulement simple d'utilisation, mais également de développement. Dans la même optique de simplicité, elle devait être facilement transportable et peu encombrante.

La solution sera composée de deux interfaces : une à destination de l'élève déscolarisé et l'autre à destination de la classe.

1) Stratégie de développement

Pour construire cette application, nous avons décidé d'utiliser une méthode serveur/client où le client envoie des informations au serveur (ex : son identifiant et son mot de passe lors de la connexion) et le serveur traite l'information et produit une réponse en conséquence.

Pour réaliser cela, une des méthodes les plus utilisées est la création d'une API REST. Le but de cette méthode est de créer des requêtes de différents types (GET-POST-PUT-DELETE) que le client doit appeler dans le front end.

Nous avons donc commencé par réfléchir aux différentes requêtes dont l'application aura besoin.

URL	GET	POST	PUT	DELETE
/api/student/all	donne la liste de tous les étudiants inscrits			supprime tous les étudiants
/api/student/{id}	donne l'étudiant de cet id	créé un nouvel étudiant avec cet id	modifie l'étudiant de cet id	supprime l'étudiant de cet id
/api/professor/all	donne la liste de tous les professeurs inscrits			supprime tous les professeurs
/api/professor/{id}	donne le professeur de cet id	créé un nouveau professeur avec cet id	modifie le professeur de cet id	supprime le professeur de cet id
/api/student/{id}/documents/all	donne tous les documents de cet étudiant			supprime tous les documents de cet étudiant
/api/professor/{id}/documents/all	donne tous les documents de ce professeur			supprime tous les documents de ce professeur
/api/student/{id}/matiere/{matiere}/documents	donne les documents de cette matière pour cet étudiant			supprime les documents de cette matière pour cet élève
/api/document/{id}	donne le document de cet id	créé un document de cet id	modifie le document de cet id	supprime le document de cet id
/api/student/{student}/avatar	donne l'avatar de cet étudiant		enregistre l'avatar de cet étudiant	

TABLEAU RECAPITULATIF DES REQUETES

Ces requêtes sont celles sur lesquelles nous travaillons au terme du projet.

2) Application côté client

Comme énoncé lors de notre précédent rapport, nous nous sommes servis de la bibliothèque JavaScript React.js afin de développer le côté client de notre application. L'implémentation de React nous a permis de créer facilement l'interface utilisateur de notre application.

En effet, le but de cette bibliothèque est de concevoir du code réutilisable, pouvant s'adapter aux différents composants de notre projet. L'architecture de notre solution regroupe donc des composants HTML/CSS et JS.

Cependant, comme évoqué dans le rapport du 1er semestre, ce ne sont pas les seuls avantages de React.

Il s'agit d'une bibliothèque très populaire, qui possède une documentation riche, alimentée et régulièrement mise à jour par une communauté active. Nous nous sommes d'ailleurs aidés de nombreuses chaînes YouTube, newsletters, blogs... afin de concevoir l'interface client de cette application ou bien de résoudre un problème au niveau du code. Effectivement, il est quasiment impossible que personne n'ait déjà rencontré le même problème.

De plus, il nous a paru intéressant de mettre en pratique les procédés de code actuels qui sont enseignés à ISIS.

Concernant la hiérarchie globale des dossiers côté client, nous avons décidé de procéder comme ceci :

Tout d'abord, nous avons préféré créer deux sous-dossiers "mimi-back" et "mimi-front" afin de distinguer proprement le côté serveur du côté client.

Du côté client, se trouve premièrement un dossier "public" comprenant lui-même un dossier "img" car nous avons eu besoin d'importer certaines images pour construire notre application.

Les dossiers assets, componentsDragDrop et config, nous ont servi à la création d'un dépôt de fichiers de type "glisser déposer". Ce dépôt est un outil utilisé par les professeurs pour envoyer les cours aux élèves.

Pour concevoir cet outil, nous nous sommes aidés d'une vidéo YouTube (Annexe 1).

Nous avons également un premier dossier components, qui contient lui-même un dossier "navbar" ainsi qu'un fichier Logo.js. Ce dossier nous est utile afin de regrouper les différents composants réutilisables de notre application. En effet, un composant à la racine du dossier components est réutilisable dans toute l'application, ainsi on constate qu'il y a une NavBar par page.

Le dossier "pages" regroupe tous les fichiers.js servant à la construction de l'application.

Concernant le design de nos pages, nous avons décidé de créer un dossier "styles" dans lequel se trouve un fichier _settings.scss permettant de définir toutes les modifications CSS communes. En effet, la plupart des pages possédant quasiment la même structure, il nous a paru plus judicieux de créer ce fichier en regroupant les éléments dont le CSS est identique sur chaque page.

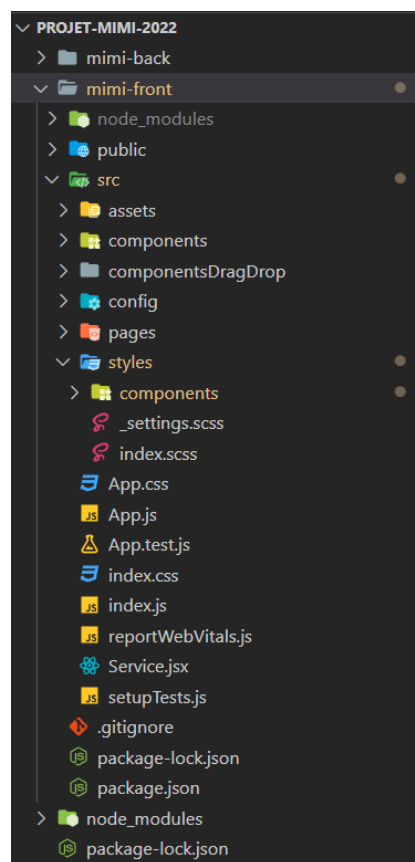
Le dossier components contient des fichiers.scss permettant de réaliser le design des pages de notre application. SCSS est le sur-ensemble de CSS. SCSS est la version la plus avancée de CSS.

Le fichier index.scss nous permet quant à lui d'importer tous les fichiers.scss.

Le fichier App.js sert à centraliser les différentes pages de notre application ainsi qu'à décrire les routes de ces pages. Effectivement, les pages peuvent être reliées les unes aux autres en cliquant sur des boutons par exemple.

Service.jsx est un fichier qui nous permet d'écrire les requêtes permettant de récupérer ou de modifier les informations de la base de données.

Enfin, certains dossiers comme les dossiers node_modules, .gitignore, README.md, package.json, package-lock.json, index.js... sont créés de base lors de l'implémentation de React.



STRUCTURE DU FRONT END

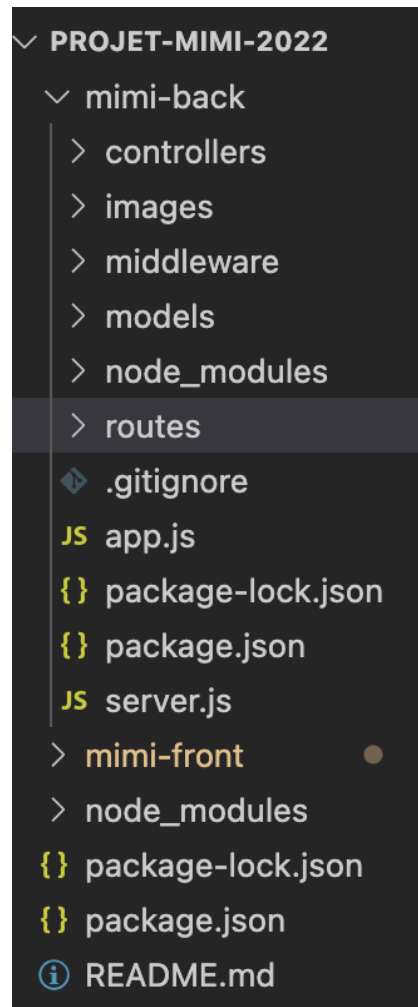
3) Application côté serveur

Pour le back end de ce projet, nous avons décidé d'utiliser Node.js qui permet d'utiliser le langage de programmation Javascript pour développer un serveur.

Nous avons en effet pour consigne de ne pas utiliser Java et son Framework Spring, qui est le langage que nous pratiquons le plus à ISIS.

Nous avons cependant trouvé pertinent d'utiliser Node.js car c'est un logiciel libre, qui possède une grande communauté et qui permet de créer des applications très efficaces.

De plus, le front end étant développé en React, qui est également une librairie Javascript, nous pouvons nous aider dans la programmation.



STRUCTURE DU BACK END

Nous avons décidé de structurer notre back end comme cela :

- Les controllers servent à définir les requêtes de l'API REST. Nous y écrivons le code qui explique ce qu'il doit se passer lors d'une inscription, d'une authentification ou de l'ajout d'un document par exemple.
- Nous avons créé un middleware qui permet de créer des tokens lors de l'inscription et la connexion d'un utilisateur. Ces tokens permettent de vérifier que l'utilisateur qui navigue sur l'application accède bien aux données qu'il a le droit de voir et qu'il ne fasse pas des choses qu'il n'est pas supposé faire. Un élève n'aura par exemple pas les mêmes droits qu'un professeur.
- Les modèles sont les classes qui permettent de créer nos objets Javascript. Ce sont ces objets que l'on appelle ensuite dans les requêtes REST (User, Student, ...). Nous avons créé ces modèles avec mongoose. Nous avons également utilisé la librairie unique-validator pour vérifier notamment qu'une adresse mail ne peut être utilisée que par un utilisateur.
- Enfin, les routes permettent d'indiquer au serveur la structure de l'API, on y indique la route et la méthode. On appelle alors les fonctions créées dans les controllers.

Pour ce qui est de la base de données, nous avons décidé d'utiliser MongoDB, qui est NoSQL et est très utilisée en Node.js.

La création et la connexion à la base de données a été réussie, et nous pouvons bien créer des objets et les stocker dans la base.

Nous avons cependant rencontré beaucoup de difficultés lors du développement de ce serveur.

Il nous a fallu apprendre une nouvelle méthode de programmation de O, nous avons donc décidé de dédier une dizaine d'heures sur une formation en Node.js pour deux membres du groupe.

Malgré cette formation en ligne, il est très difficile d'avancer dans le développement sans quelqu'un à qui poser des questions. Nous nous retrouvions alors souvent bloqués dans l'avancement à cause d'un manque de connaissances.

Nous sommes néanmoins satisfaits du résultat produit car nous avons réussi à initier un serveur web, connecté à une base de données et qui est prêt à recevoir des requêtes de l'API REST.

Description de la solution

Les pages de l'application sont toutes au format tablette puisque le but, à terme, est de rendre disponible ce dispositif sur une tablette fournie par l'ADPEP. La taille des différents boutons, icônes... ont été calibrés en conséquence pour rendre la solution pratique (boutons format tablette plus gros).

Vous pourrez retrouver toutes les illustrations de l'application prises en capture d'écran en annexes.

Voici un résumé de l'utilité et de l'architecture des différentes pages :

Connexion :

Lorsqu'il s'agit de la première connexion à l'application, il est possible de s'inscrire en cliquant sur le lien "Pas de compte ? Inscrivez-vous". Des informations sont par la suite demandées concernant l'établissement, l'élève et le parent responsable. À la fin de cette inscription, un message de confirmation s'affiche et les futurs utilisateurs seront prochainement contactés par l'ADPEP via mail.

Une fois inscrit, rien de plus simple pour se connecter à notre solution. Il vous suffit de rentrer votre identifiant / mot de passe.

Le lien "Qui sommes-nous ?" permet aux potentiels futurs utilisateurs de se renseigner à propos de l'ADPEP.

Côté enfant :

Le but premier de notre application étant de permettre à des enfants déscolarisés de suivre les cours à distance, sans perte de lien social. Il nous a paru cohérent d'essayer de créer une application avec un design assez enfantin et simple d'utilisation. Ainsi, toutes les pages destinées à être utilisées par les enfants sont stylisées en conséquence. Par exemple, la police de la NavBar ressemble à du texte manuscrit, ce qui ne ressemble pas aux polices des applications en temps normal. Nous avons également utilisé de nombreuses images/logo pour que cela soit plus visuel et attractif (comme la page des jeux).

Au niveau des fonctionnalités disponibles pour ce type d'utilisateur, nous retrouvons les fonctionnalités citées en amont dans ce rapport. Elles sont disponibles via le "menuEleve".

Ainsi, une page de cours permet à l'élève de visualiser les cours envoyés par ses professeurs en fonction de leurs matières d'enseignement. Une page visioconférence lui permet d'assister en temps réel et d'intervenir pendant les cours à distance (fonctionnalité à développer). L'élève a par ailleurs accès à une page où il pourra sélectionner l'avatar qu'il a envie d'utiliser pour se représenter en cliquant sur les icônes. Enfin, l'enfant pourra jouer avec ses camarades de classe grâce à des jeux en ligne disponibles dans la page "jeux" en cliquant sur les logos.

Côté professeur :

Le design de l'application change quelque peu pour les professeurs puisque nous n'avons pas eu besoin de donner cet aspect enfantin.

Une fois le professeur authentifié, la page “menuProf” lui permet de choisir les fonctionnalités qu’il veut exécuter.

À terme, une fonctionnalité de visioconférence sera disponible pour permettre à l’enseignant d’assurer son cours aux élèves qui se trouvent à distance. Le professeur en question pourra également déposer son support de cours et le partager aux enfants utilisant la plateforme grâce à l’option “déposer des cours”.

Fonctionnalités restantes à développer

1) La visioconférence

L'une des fonctionnalités principales de notre solution est la visioconférence. En effet, au travers de celle-ci, les élèves pourront suivre les cours en distanciel, intervenir à tout moment pour poser des questions (grâce à un système d'emojis visible par l'enseignant) et interagir avec leurs camarades à l'aide d'une fonctionnalité "chuchotement".

Malheureusement, nous n'avons pas eu le temps de créer notre propre système de visioconférence et ainsi à l'intégrer au sein de l'application. Il s'agit d'un des paramètres de l'application le plus difficile à développer selon nous. En effet, nous nous sommes quelque peu renseignés à ce sujet et nous avons trouvé une vidéo Youtube expliquant comment intégrer un système de chat vidéo à l'aide de React et d'une plateforme nommé "Agora.io" (lien de la vidéo en Annexe 2).

Agora.io est une plate-forme de développement qui fournit des appels de diffusion, vocaux et vidéo pour les applications mobiles et Web via leur kit de développement logiciel (SDK).

Cependant, nous avons connaissance du prolongement de notre PTUT, en stage de 4^e année, et nous pensons qu'en suivant les indications de cette vidéo, il sera possible de créer un système d'appel vidéo. Restera encore à trouver comment implémenter toutes les autres fonctionnalités (demander la parole pour intervenir, chuchoter avec ses camarades, envoyer des messages via un chat...).

Nous avons donc choisi une alternative au développement complet d'un système de visioconférence, car nous avons donné la possibilité à l'élève de se connecter à la plateforme Zoom à l'aide d'un lien. Ce lien est disponible dans la page visioconférence du menu élève. Même si cette alternative n'est pas optimale, puisqu'elle ne permet pas à l'enfant déscolarisé d'accéder à toutes les fonctionnalités que nous aurions aimé mettre en place pour maintenir un certain lien social.

2) Relier le front end et le back end

Nous n'avons pas pu relier le back et le front de notre application. En effet, cela aurait peut-être été plus simple de réaliser la construction du back end avec le Framework Spring de Java étant donné que nous avons pu l'utiliser lors de divers projets à ISIS.

Cependant, pour remédier à ce problème, il faudrait, dans le fichier Service.jsx du côté client, écrire toutes les requêtes permettant de récupérer ou de modifier les informations de la base de données (à l'aide des méthodes GET, DELETE, PUT, et POST). Requêtes indispensables pour faire des appels à l'API REST créée dans le back.

Nous avons tout de même tenté de réaliser ce lien, en se servant d'une dépendance appelée axios (tutoriel en Annexe 4). Axios nous permet simplement de faire des requêtes HTTP à notre back end. Mais malheureusement, cela n'a pas fonctionné.

3) Développement d'un jeu multijoueur

Nous avons tenté, à l'aide des nombreuses ressources existantes concernant le développement React, de développer un jeu multijoueur en local directement disponible via notre application. Nous avons trouvé de multiples vidéos à ce sujet, mais malgré cela nous n'avons pas réussi à les implémenter au projet.

Tout d'abord, nous nous sommes vite résolus à l'idée de développer un jeu multi-joueurs, car même s'il s'agit d'une fonctionnalité essentielle au maintien du lien social, cela s'est avéré très compliqué à réaliser. Nous avons donc essayé de développer localement un jeu "simple" tel que le jeu du pendu, un pierre/feuille/ciseaux, un jeu du Snake... mais nous n'avons pas réussi à les adapter à notre solution.

Nous avons donc réfléchi à une solution de substitution, en implémentant des liens vers des jeux multi-joueurs en ligne. Ainsi, les élèves en distanciel pourraient tout de même jouer en ligne avec leurs camarades de classe.

4) Choix de l'avatar

Au sein de l'application, il devrait être possible que l'élève puisse choisir un avatar différent en fonction de ses envies. Ce choix devrait être stocké en mémoire, même lors de la déconnexion de la session, et ce, jusqu'au nouveau changement de profil et ainsi de suite.

Comme nous n'avons pas réussi à relier le front et le back et à réaliser l'authentification, l'élève ne peut pas encore accéder à cette fonctionnalité. À ce stade, l'élève peut seulement visualiser les types d'avatars mis à disposition. Il reste encore beaucoup de travail de ce côté-là. En effet, l'enfant doit pouvoir, dans l'idéal, choisir une photo de sa galerie pour la définir comme photo de profil, ou bien avoir le choix entre plusieurs Mii/avatars personnalisables, ou encore choisir de montrer son visage grâce à l'option "facecam".

Il s'agit d'un choix de développement de notre part de ne pas avoir priorisé toutes ces fonctionnalités, car cela nous a semblé moins important à réaliser en premier lieu, étant donné la charge de travail à réaliser par rapport au temps imparti.

5) Stockage des fichiers

Nous avons réussi à implémenter un système de "glisser-déposer" afin que les professeurs puissent déposer les cours des étudiants concernés par le dispositif. Néanmoins, nous n'avons pas réussi à relier le front et le back de notre application pour que ces cours soient stockés dans une base de données.

Pour l'instant, le professeur peut juste faire glisser des fichiers, mais ils ne seront disponibles nulle part et donc pas visible par les élèves.

Pour résoudre ce problème, il faudrait procéder comme énoncé ci-dessus dans la partie 2 pour relier le front et le back de notre solution.

Conclusion

Ce projet a été très enrichissant puisque nous avons dû mettre en place une démarche très progressive et organisée. La réalisation d'une telle solution implique une réelle collaboration entre les différents membres du groupe. Ce qui nous a permis d'exploiter nos compétences respectives en les mettant en commun. Ce partage de connaissances a donc été bénéfique à l'ensemble du groupe, car nous avons pu découvrir de nouveaux outils, tels que MongoDB, Express, Node.js ... tout en nous familiarisant à des technologies étudiées en classe comme React.

Cependant, durant tout ce projet, nous avons eu de nombreux problèmes auxquels il a fallu remédier. Cela nous a demandé un effort de recherche, car nous avons dû nous documenter sur les différents outils par nous-mêmes. Un des principaux problèmes a été l'utilisation de Node.js. Effectivement, nous avons perdu beaucoup de temps en nous formant à cet outil back end et nous avons eu quelques difficultés pour l'utiliser.

L'analyse de la réalisation de nos objectifs disponible dans ce rapport nous permet de nous situer dans notre démarche. Lors d'un futur travail de groupe, nous essaierons d'améliorer notre efficacité afin de pouvoir atteindre la totalité des objectifs fixés.

Nous avons connaissance de la poursuite de ce projet en stage et nous avons essayé d'aiguiller au maximum les futurs étudiants en charge concernant les procédés à réaliser pour développer toutes les fonctionnalités de l'application.

Bibliographie

<https://www.lespep81.fr/>

[Accompagnement pédagogique à domicile à l'hôpital ou à l'école - APADHE \(ih2ef.gouv.fr\)](#)

<https://www.atlassian.com/fr/software/jira/features>

<https://fr.reactjs.org/docs/getting-started.html>

[https://developer.mozilla.org/fr/docs/Learn/Tools and testing/Clientside JavaScript frameworks/React getting started](https://developer.mozilla.org/fr/docs/Learn/Tools_and_testing/Clientside_JavaScript_frameworks/React_getting_started)

<https://bulma.io/>

<https://openclassrooms.com/fr/courses/6390246-passez-au-full-stack-avec-node-js-express-et-mongodb>

Rapport de stage de fin d'étude de 5^e année de Rayan Barbara

Rapport PTUT Auriane Chelle - Etienne Deux

Annexes

Annexe 1 :

<https://gitlab.com/LucasAbitbol/projet-mimi-2022.git>

Annexe 2 :

<https://www.youtube.com/watch?v=IUrWJVCCVGc>

Annexe 3 :

[React Drag Drop File Input Component | React Drag And Drop | ReactJS Tutorial](#)

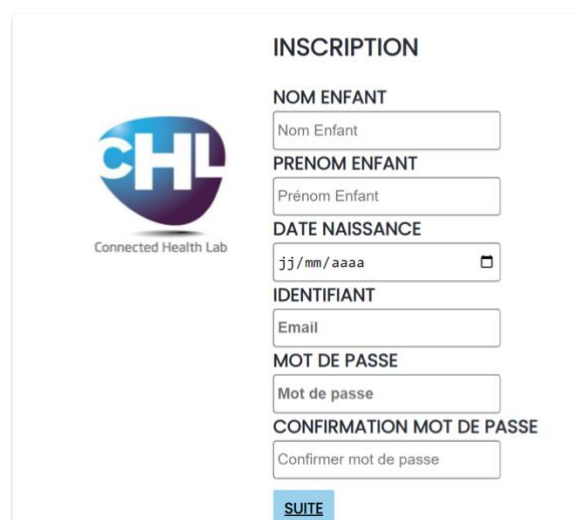
Annexe 4 :

<https://medium.com/zero-equals-false/how-to-connect-a-react-front-end-with-node-js-bccb1fb7e2bb>

Screens de l'application :



The authentication screen features the Connected Health Lab (CHL) logo on the left. On the right, under the heading 'AUTHENTIFICATION', there are two input fields: 'IDENTIFIANT' (Email) and 'MOT DE PASSE' (Mot de passe). Below these is a blue 'SE CONNECTER' button. At the bottom, there is a link that reads 'Pas de compte ? Inscrivez-vous / Qui sommes nous ?'.



The registration screen features the Connected Health Lab (CHL) logo on the left. On the right, under the heading 'INSCRIPTION', there are several input fields: 'NOM ENFANT' (Nom Enfant), 'PRENOM ENFANT' (Prénom Enfant), 'DATE NAISSANCE' (jj/mm/aaaa with a calendar icon), 'IDENTIFIANT' (Email), 'MOT DE PASSE' (Mot de passe), and 'CONFIRMATION MOT DE PASSE' (Confirmer mot de passe). At the bottom, there is a blue 'SUITE' button.



QUI SOMMES NOUS ?

L'AdPep sert à aider les étudiants de primaire jusqu'au lycée qui ne peuvent se rendre en cours physiquement.

Le but de cette application est d'offrir aux étudiants un dispositif personnalisé Afin qu'ils se sentent mieux dans le cadre scolaire.

N'hésitez pas à nous contacter au 06 13 13 13 13 pour plus de renseignements.
Notre site Web [ICI](#)



Cours Visioconférence Avatar Jeux

[Se déconnecter](#)



ACCEDER AUX COURS



VISIOCONFERENCE



CHOISIR SON AVATAR



JOUER !



Cours Visioconférence Avatar Jeux

Se déconnecter



MATHEMATIQUES



FRANCAIS



SVT



PHYSIQUE



ANGLAIS



ESPAGNOL



HISTOIRE



CHIMIE



Cours Visioconférence Avatar Jeux

Se déconnecter



Mii



PHOTO



AVATAR



FACECAM



Veuillez déposer vos
fichiers



Zone de dépôt