LES TROPHÉES NSI

Édition 2022

DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET

Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page **Préparer votre participation**.

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à **info@trophees-nsi.fr**.

NOM DU PROJET : *Schedule*

> PRÉSENTATION GÉNÉRALE :

L'objectif est de créer une plateforme sur laquelle les élèves pourront voir en temps réel quels cours ils vont avoir aujourd'hui ou cette semaine et dans quelle salle ils seront.

Dans notre lycée, il faut aller voir le tableau des remplacements tous les matins pour voir les créneaux remplacés et donc savoir qui nous surveille et dans quelle salle nous devons aller. D'autant plus que c'est le matin que nous découvrons si nous n'avons pas cours. Alors l'idée nous est venue de faciliter ce système. Ainsi les élèves et les professeurs pourront consulter les cours qu'ils ont à tout moment, et ce avec les remplacements.

L'administrateur est le seul à pouvoir modifier la base de données. Lorsqu'un professeur demande une absence, il téléphone à l'administrateur et l'administrateur ajoute l'absence dans la base de données ainsi que le professeur ou surveillant qui va le remplacer et dans quelle salle.

Il est inutile de se créer un compte, chaque utilisateur en possède déjà un.

> ORGANISATION DU TRAVAIL :

Présentation de l'équipe

Notre équipe est composé de :

- Romain LEDDA, Product Owner/ Chef de projet et développeur backend
- Daniel TESSIER, développeur full-stack
- Sacha RAMSTEIN, développeur full-stack
- Romain SCHMITZ, développeur frontend
- Alexandre LATUNER, développeur frontend

Répartitions des tâches

Nous avons été initiés à la méthode *AGILE*, très utilisée dans le monde de l'entreprise.

Ainsi nous avons utilisé la méthode *MoSCoW* qui est un principe de priorisation des tâches selon ces catégories :

- Mo : *Must have this*, ce qui est vital, le plus important
- S : *Should have this if at all possible*, ce qui est essentiel, ce qu'il faudrait au minimum
- Co : *Could have this if it does not affect anything else,* ce qu'il serait bien d'avoir mais il faut que ça n'ait aucun impact sur les autres tâches, ce qui touche au confort d'utilisation ou au visuel
- W : *Won't have this time but would like in the future,* ce qui pourrait être améliorer ou ajouté mais à faire uniquement s'il reste du temps et du budget

Chacun a pu donc choisir les tâches qu'il souhaitait réaliser en fonction de ses appétences et de ses envies selon ce tableau :

Mo	*S*	*Co*	*W*
- Développer la page de connexion - Développer la page sur laquelle l'utilisateur peut visionner son emploi du temps - Développer la page admin - Paramétrer le serveur - Agrémenter la base de données - *Ajouter la fonctionnalité* : Connexion entre la page de connexion et la page de compte principale grâce à un identifiant et un mot de passe - Se déconnecter de son compte - Afficher le tableau affichant les matières de l'utilisateur	- Entrer une absence - Remplacer l'heure d'un prof absent par étude	- Possibilité d'ajouter des items dans la BDD à partir de la page admin - Possibilité de supprimer des items dans la BDD à partir de la page admin - Ajouter un menu déroulant sur la bannière du site - Afficher le tableau affichant les absences, les profs et les élèves sur la page admin - Afficher les matières avec une couleur différente - *Ajouter la fonctionnalité* : Assignation d'un prof disponible sur un cours où un prof est absent - Modifier une donnée dans la BDD	- Développer l'application mobile - Afficher le nombre d'absence d'un professeur - Ajouter des items dans la BDD avec un fichier - Possibilité d'envoyer un message pour prévenir d'une absence via la plateforme

Nous faisions des petites réunions, *Daily Scrum*, en début de séance pour suivre l'avancement par rapport au *Sprint* et ce que nous allions réaliser durant la séance. Nous faisions aussi une grosse réunion, Weekly Scrum, avec notre professeur, chaque mercredi soit au début du nouveau *Sprint,* durant laquelle le *Product Owner* informait le professeur de l'avancement du projet et du travail qui allait être réalisé pour le prochain *Sprint*. Un *Sprint* correspond à une semaine

Pour s'organiser, nous avons utilisé *Notion, Git* et *GitHub, la messagerie d'EcoleDirecte* ainsi qu'un groupe *Discord* en dehors du cadre scolaire.

Sur le *Notion*, les tâches étaient réparties sur différentes colonnes selon ce schéma :

To Do	Sprint	Doing	Done	Secondary

La colonne *To Do* contient l'intégralité des tâches à réaliser. La colonne Sprint contient toutes les tâches à réaliser durant le Sprint. La colonne *Doing* contient toutes les tâches en cours et *Done* toutes les tâches réalisées. La colonne *Secondary* contient les tâches dont on s'est rendu compte pendant le projet qu'elles n'étaient pas indispensables. Les membres du groupe

peuvent prendre la tâche qu'ils souhaitent en fonction de leurs compétences ou du domaine d'apprentissage qu'ils souhaitent étendre et mettre leur nom dessus ainsi que la date où il a commencé à travailler dessus. Les membres peuvent donc voir en temps réel quelle tâche est à prendre et quelle tâche a été complétée ce qui est bien pratique. Le Product Owner supervise ce système et assigne des tâches quand personne ne les a prises.

LES ÉTAPES DU PROJET :

Nous avons eu 5 semaines donc 5 *Sprints* :

- Idée du projet + Recherches
- Prototypage de la plateforme
- Elaboration du design du site
- Réalisation des différentes tâches
- Test
- Finalisation

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Ce qui est terminé :

- Le serveur
- L'architecture des pages
- La base de données
- Interaction entre la BDD et le serveur
- Possibilité de se connecter
- Possibilité de se déconnecter
- Possibilité de voir ses cours
- Ajouter des absences dans la BDD
- Ajouter un remplaçant
- Ajouter, supprimer et modifier des items dans la BDD
- Les emplois du temps des élèves sont modifiés avec les remplacements
- Les élèves peuvent voir les cours avec les remplacements
- Les matières ont une couleur différente

Pour vérifier l'ergonomie et l'esthétique de la plateforme, nous avons recueillis des avis auprès d'autres camarades ainsi que notre professeur et visionner des vidéos YouTube ou des modèles d'UX design.

Pour tester et *debugger* notre projet, nous nous sommes mis à la place des différents utilisateurs pour voir ce qui fonctionnait, si on ne rencontrait pas de problèmes..etc. Nous avons testé différentes données parfois volontairement incompatibles.

Difficultés rencontrées et solutions apportées

- Comprendre l'architecture d'un site, les interactions entre le frontend et le backend ainsi qu'entre le serveur et la base de données, page statique et dynamique, les requêtes http, le fonctionnement des technologies qu'on utilisait (Node.js, Express, Fetch…), le système de *Promesses…*
- Les méthodes d'importation de modules JS
- Erreur de parsage. Solution : Installer des modules pour parser les données.
- Oubli de certains cas dans la gestion d'erreur et des différentes situations.
- Incompatibilité de certaines données.
- Difficulté à s'approprier les différentes technologies.

> OUVERTURE :

Idées d'améliorations

- Héberger le site web
- Développer une application web. L'application permet aux élèves et professeurs de visualiser leur emploi du temps. De plus, elle envoie une notification chaque matin, à chaque professeur pour savoir s'ils seront là durant la journée et modifier l'emploi du temps des élève en conséquence.
- Améliorer l'ergonomie et l'esthétique de *Schedule*
- Possibilité d'envoyer un message à l'administrateur via la plateforme pour prévenir d'une absence
- Ajouter un message d'erreur si l'utilisateur entre un mauvais identifiant.
- Responsive design

Stratégie de diffusion

- Tout d'abord, nous allons le présenter à notre chef d'établissement pour savoir s'il serait intéressé pour l'intégrer dans le système administratif de notre lycée. Une présentation à l'ensemble des élèves de 6eme de notre établissement est prévue le 3 juin (au cours d'une après-midi dédié à la présentation des différents projets).
- Envoyer des mails ou démarcher téléphoniquement des lycées dans notre secteur afin d'avoir un avis concret pour améliorer *Schedule* voire trouver des clients.

Analyse critique du résultat

L'organisation était très efficace dans l'ensemble. Chaque membre était indépendant et impliqué. Il y a eu à un moment donnée une petite baisse de productivité due à la période. La liberté laissée à chacun et la solidarité entre les membres ont permis d'atteindre ce résultat. Si on possédait les connaissances au début du projet, on aurait été d'autant plus productif et efficace.

DOCUMENTATION

Guide d'utilisation

Personnellement, nous avons utilisé l'IDE *Visual Studio Code* mais vous pouvez utiliser celui que vous voulez :

- Installer Node.js via ce lien
- Décompresser l'archive
- Dans un terminal (invite de commande), se placer dans le répertoire principal du projet :
  cd schedule_project (ou le répertoire que vous avez choisi)
- Installer le module Express : *npm install express*
- Installer le module Handlebars: *npm install handlebars*
- Installer le module Sqlite3 : *npm install sqlite3*
- Lancer le serveur : *node server.js*
- Ouvrir un navigateur internet et entrer l'url suivante : *localhost:8000*
- Pour se connecter veuillez voir ci-dessous
- Pour arrêter le serveur : *CTRL + C*

Pour tester la page d'un élève, utiliser :

	Classe 1	Classe 2	Classe 3
Identifiant	- M.Petitjean - J.Leroy	- B.Fontaine - S.Tessier	- J.Schneider - D.Navarro
Mot de passe	*1234567890*		

Pour tester la page d'un professeur, utiliser :

Identifiant	J.Parmentier	D. Tessier
Mot de passe	1234	1234

Pour tester la page de l'administrateur, utiliser :

- *S.Moebel*
- *Nsilove2004*

Il faut faire attention à la casse !

Organisation du site

- Page de connexion : Lorsque les utilisateurs entrent l'URL du site, ils tombent sur une page de connexion. Ils doivent entrer l'identifiant et le mot de passe qui leur a été assigné pour se connecter à leur compte
- Page de compte principale : Une page qui sur laquelle les utilisateurs peuvent visualiser leur emploi du temps. En haut à droite, l'utilisateur peut se déconnecter.
- Page admin : La page réservée à l'administrateur. Il peut voir les profs absents qu'il a entré dans la base de données de l'établissement. Il peut voir les professeurs de l'établissement et ajouter des professeurs. De même pour les élèves. Il peut également remplacer un prof absent par un autre prof disponible à l'heure du cours en question. Il peut également se déconnecter grâce à l'icone en haut à droite.

Technologies et bibliothèques utilisées

- Node.js : Node.js est un *run-time* Javascript, c'est-à-dire un environnement d'exécution Javascript. Il permet d'exécuter du code JS en dehors d'une page web donc est beaucoup utilisé pour la partie *backend* d'un projet. Il a la particularité d'être mono-thread et asynchrone. Mono-thread signifie que qu'il n'y qu'une file d'exécution par tâche et asynchrone qu'il n'a pas besoin qu'une ligne d'instruction soit terminée pour passer à la suivante ce qui est bien pratique. Sinon dans le cas d'un serveur ce serait ingérable.
- Express : Express est un framework Node.js qui fournit des applications web fondamentales comme l'hébergement d'une page web statique ou la récupération de requêtes. Il fonctionne sous la forme d'une suite de *middlewares :* Une fonction prenant en paramètre une requête et une réponse.
- Handlebars : Handlebars est un moteur de templating en JS qui permet de modifier une page web sur un serveur et de l'afficher quand l'utilisateur en fait la requête.
- Sqlite3 : Sqlite3 est un module JS qui sert d'interface entre la bibliothèque SQLite utilisée en cours et Node.js