Pré-Ing 2 MI Groupe 6



par Nathan Fouchet, Émilien Massi, Xavier Dupontrouvé et Camille Laborde

Rapport du projet Web

Sommaire :

* Conception **3**
  + Par Nathan**3**
  + Par Émilien**4**
  + Par Xavier**5**
  + Par Camille**6**
* Analyse du projet**6**
  + Planning**6**
  + Défauts du site**7**
  + Bonus réalisés**7**
* Manuel d’utilisation**8**
* Manuel technique**9**
* Bibliographie**10**

Conception

Par Nathan :

Par Émilien :

Il existe une partie du code importante que personne d'autre voulait faire dans ce projet, je parle du **CSS**. Avec l'accord de mon équipe, je suis parti sur l'esthétique du projet. Ma contribution a été très **séparée** des autres puisque personne n'avait directement besoin de mon travail et rien n'avait de rapport avec le reste du projet. Cela étant dit, je me suis mis au travail dès que possible sachant que j'allais avoir besoin de temps et de patience, le moindre problème pouvait vite dégénérer avec tout le reste lorsqu'on écrit en CSS.

Un bon exemple de petit problème devenant un gros problème est l'histoire du **dégradé**. Pour le contexte, le bandeau était fini mais je voulais le rendre un peu meilleur en m'inspirant des exemples sur la consigne du projet. C'est là que j'ai eu l'idée de mettre un dégradé de gris sur le bandeau. Au bout de 2 cours entier dessus soit 6 heures, j'ai dû abandonner cette partie. Le principal souci vient du logo faisant aussi partie du "fond" du bandeau et donc rentrait en conflit avec le dégradé qui était considéré comme une image. Ce problème était devenu aussi complexe que les **mails** de Nathan.

Après cet incident, la seconde partie du CSS était la **sélection**, c'est-à-dire les quatre boutons liant les différents modules du projet. Je me suis amusé sur cette partie en découvrant les animations avec **padding**. Ces petits boutons sont remplis de petits détails pour les rendre plus ergonomiques et esthétiques. Par exemple, l'espacement entre les différents mots ne sont pas copiés-collés sur les quatre, chacun a son espacement pour que l'animation soit équilibrée entre chaque bouton. Les boutons deviennent noirs si on clique pour bien montrer que le clic a été pris en compte (si on lag).

Enfin, pour avoir un beau site, il faut un beau **fond** ! J'ai utilisé la même image que sur les exemples et j'ai choisis un blanc un tout petit peu plus sombre pour ne pas faire mal aux yeux (c'est vraiment minime). Le seul problème sur ce point était que je n’avais prévu qu'un seul fichier CSS pour tout le projet mais je ne savais pas comment faire pour avoir un fond différent pour chaque site depuis le même fichier CSS. Internet ne m'a pas aidé sur ça donc j'ai dû me résoudre à créer un fichier par fond. C'est pourquoi il y a un fichier CSS par module ce qui, après coup, a été utile pour utiliser le même id pour deux couleurs différentes par exemple.

Pour finir, nous nous sommes préparés à **l'oral**. Par Xavier :

Comme convenu avec mon équipe, j’étais responsable de réaliser le code concernant l’inscription et la connexion de l’utilisateur sur le site. Je n’ai malheureusement pas été très actif durant les premières semaines du projet à cause de mes **problèmes de santé** qui m’ont fait m’absenter de nombreuses fois.

Pour **l’inscription** j’ai réalisé une page en html qui affiche des informations avec des champs à remplir. Les informations écrites sont transmises par le Javascript (requête AJAX) au PHP qui utilise plusieurs fonctions qui vérifient les informations données par l’utilisateur. J’ai rencontré des difficultés à faire transmettre les informations par la **requête AJAX** avec la méthode POST. Ce qui m’a posé problème était le placement de certaines lignes dans le javascript qui lui a causé d’ignorer des instructions importantes.

Le PHP aussi m’a donnée pas mal de fil à retordre, en effet la fonction servant à vérifier **le format de la date** de naissance de l’utilisateur a été particulièrement difficile à mettre en place. Je devais diviser la chaîne de caractères avec les « / ». C’est après de nombreuses recherches sur internet que j’ai réussi la fonction.

Enfin pour **créer le fichier** « user.txt » avec les informations de l’utilisateur (lors de la création d’un compte) je devais trouver un moyen pour générer le nom du dossier de chaque nouvel utilisateur pour qu’ils n’aient pas le même nom de dossier. Mes camarades m’ont aidé à utiliser une fonction permettant de générer le nom de dossier pour les nouveaux utilisateurs.

Après voir fini le système d’inscription je me suis chargé de ce qui suivait, la **connexion** et la **page d’accueil de l’élève**.La connexion était assez simple à réaliser, j’ai repris beaucoup de choses du système d’inscription. Cependant il m’a été difficile de concevoir la fonction qui sert à **vérifier l’email et le mot de passe** en même temps écrit par l’utilisateur lors de sa connexion.

Suite à cela j’ai réalisé la **page d’accueil de l’élève**, avec les boutons qui nous renvoient sur plusieurs autres pages. Je me suis occupé de l’option **modification du profil** étant donné que c’était moi qui avais fait la création du profil. Je n’avais besoin que de récupérer encore une fois du système de création de compte quelques fonctions mais en faisant en sorte que les données remplacent celles du fichier de l’utilisateur connecté.

Pour la suite, nous nous sommes **préparés pour l’oral**.

Par Camille :

Étant en **vacances**, et sans ordinateur, je n’ai pu que coder la partie html du module consultant (entre le … et le …), qui m’a permis de mieux découvrir la consigne.

En rentrant, j’ai essayé d’installer **Wamp** sur mon ordinateur personnel. L’application ne fonctionnait pas. En suivant les conseils de Nathan et Émilien, j’ai téléchargé **Xamp** qui m’a permis par la suite de coder à la maison.

J’ai **créé le lien GitHub** qui contenait une branche par module (en plus de la branche « *main* »), et invité les membres du groupe à y participer. J’ai également créé un serveur Discord pour faciliter la communication au sein du groupe.

Pour coder la partie ***« Inclusion des références validées dans le CV »*** du module étudiant, j’ai choisi d’utiliser le langage PHP pour afficher les données sauvegardées sur le serveur puis du Javascript pour les mettre en forme (en format HTML ou en format PDF) comme demandé par l’utilisateur.

La **mise au format HTML** fut la plus simple. Pour la **mise au format PDF**, j’ai très vite trouvé plusieurs bibliothèques qui me permettaient d’en sauvegarder à partir du Javascript. J’ai essayé la bibliothèque jsPDF qui me semblaient être la plus intuitive. Seulement, elle ne me permettait pas à elle seule de créer un tableau dans mon PDF. En effet, il me fallait rajouter la bibliothèque Autotable. Après plusieurs essais, je n’ai pas réussi à manipuler les paramètres de celle-ci. Cette bibliothèque me renvoyait toujours la même mise en forme. J’ai trouvé peu de documentation qui aurait pu m’apprendre à m’en servir. Après avoir demandé conseil au professeur (Monsieur Galouz), j’ai recherché à créer un PDF à partir du PHP. Solution qui s’est avérée plus compliquée encore (inclusion de bibliothèques plus difficile). Lassée par toutes ces recherches qui n’ont portés leurs fruits, j’ai choisi d’écrire mon tableau avec un système de **coordonnées**. Cette solution m’a posé de nombreux problèmes. Le plus difficile était de gérer le tableau de savoirs-être au sein de chaque tableau référence. Pour le système de coordonnées, il m’a fallu une fonction qui prévoyait la largeur d’une colonne d’un tableau, une fonction pour la largeur de chaque lettre et une autre pour la hauteur de chaque lettre. Il reste de nombreux défauts. Certains caractères ne sont toujours pas les bons, comme « *&* » qui devient « *&amp ;* » (dû à son écriture en html). La fonction qui retourne la largeur des lettres n’a été testé que sur un nombre de caractères limités, il est donc possible que certains caractères posent problèmes à la création du PDF. Une fois le système de coordonnées finis, je me suis aperçue que mon programme refusait d’ajouter des pages lorsque c’était nécessaire, il m’a donc fallu une fonction de plus.

Pour la partie ***« Envoie des références validées au consultant »*** du module étudiant, j’ai demandé conseils à Nathan qui avait déjà réalisé la partie *« Création d’une demande de référence »* du module étudiant. Puis j’ai attendu qu’il résolve le problème lié à l’envoie des mails.

Ensuite, mon **ordinateur personnel s’est cassé**. Je n’ai pu travailler chez moi pendant quelques jours. Heureusement, Émilien et Xavier m’ont tous deux proposé de récupérer un ordinateur portable qu’ils n’utilisaient pas. J’ai pu continuer de travailler chez moi, avec l’ordinateur de Xavier.

Pour la partie **Jeux de tests** du projet, j’ai essayé en classe de télécharger les logiciels nécessaires. Je voulais pouvoir demander de l’aide au professeur, n’ayant jamais réalisé ce genre de tests. Après quelques recherches, je n’ai pas trouvé de façon de télécharger les logiciels nécessaires sans la commande « *sudo* » (qui nous ai interdite au sein de l’école). J’ai demandé conseils au professeur (Monsieur Galouz), qui n’a trouvé de solution au problème et qui nous a autorisé à ne pas réaliser ces Jeux de tests.

J’ai commencé à **organiser les fichiers** de notre répertoire GitHub, fusionner les branches, supprimer les fichiers inutiles, créer des dossiers, etcétéra...

Je suis passé à une **phase de vérification** de l’ensemble du programme, et ai décidé (pour alléger le tout) de créer le fichier « *constants.php* ». Ce fichier permet de partager certaines constantes entre les fichiers. Son objectif est de faciliter la modification de celles-ci. Je voulais que ce fichier soit utilisé par les fichiers PHP comme par les fichiers Javascript. J’ai dû créer un fichier pour les constantes du Javascript à part « *constants\_js.php* ».

Pour la suite, nous nous sommes **préparés pour l’oral** et j’ai réalisé un PowerPoint.

Analyse du projet

Planning final :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Installation de Wamp/Xamp sur les ordinateurs personnels  Création et connexion au répertoire GitHub  Découverte de la consigne  Organisation collective (planning, choix de la base de données, etc…) | Xavier code la partie HTML du référent  Nathan code la partie HTML du module étudiant  Émilien code la partie HTML du visiteur  Camille code la partie HTML du module consultant | Nathan code la partie *« Création d’une demande de référence »* du module étudiant  Camille code la partie *« Inclusion des références validées dans le CV »* du module étudiant | Camille code la partie *« Envoie des références validées au consultant »* du module étudiant  Nathan code le module référent | Camille commence la rédaction du rapport | Rédaction du rapport collective  Préparation collective à l’oral  Camille organise le répertoire GitHub |
| Nathan code la partie « *envoie de mails* » | |
| Émilien code le CSS | | | |
| Xavier code la partie *« Création de compte »*, « *Connexion* » et « *Modification du profil* » du module étudiant | | | |

La répartition des tâches s’est faite en fonction des **goûts et les capacités** de chacun. Camille n’apprécie pas le CSS contrairement à Émilien à qui le langage a été confié. Nathan a des facilités en informatiques et s’est donc vu attribué des tâches plus difficiles comme l‘envoie de mails.

Xavier, ayant pris du retard (problèmes de santé) a dû confier quelques-unes de ses tâches (rédaction des messages d’erreur et des description) à d’autres personnes.

Défauts du site :

Lorsque qu’un étudiant choisit de créer un PDF avec ses références, il est possible que certains caractères posent problèmes. En effet, le programme (« *CreateRefFile.js* ») utilise un système de coordonnées qui nécessite de connaître la largeur de chacune des lettres. Malheureusement, nous connaissons la largeur d’un nombre limité de charactères.

Lorsque qu’un étudiant choisit de créer un PDF avec ses références, certains caractères ne sont pas les bons, comme « *&* » qui devient « *&amp ;* ».

Bonus réalisés :

Manuel d’utilisation

Manuel technique

Schéma de l’organisation de nos fichiers :

Légendes :

: « Contient »

: « Peut appeler »

ou « Utilise »

: Dossier

: Fichier

écrit par Xavier

écrit par Nathan

écrit par Émilien

écrit par Camille

: Utilise le dossier Data

: Appelle le fichier constants.php et constants\_js.php

Report

README.md

Rapport.docx

Consultants.html

connected.php

createRefFile.js

references.php

profile.js

profile.php

createaccount.php

createaccount.js

connect.js

connect.php

Data

user.txt

comRef001.txt

001

refNumber.txt

Ref001.txt

Remerciements\_referent.php

checkboxes.js

link.js

referent.php

request.php

constants.php

constants\_js.php

imagejeune.png

logo.png

student.css

student.html

visitor\_objectives.php

visitor\_partners.php

Instructions

Enonce.pdf

Jeunes 6.4.pdf

Bibliographie

* Recherches sur les bibliothèques qui permettent de créer un PDF en Javascript : <https://medium.com/coderbyte/generate-a-pdf-with-javascript-3e53ca7b47e>
* Recherches sur la bibliothèque jsPDf (js) : <https://stackoverflow.com/questions/16858954/how-to-properly-use-jspdf-library>
* Recherches sur les bibliothèques jsPDF Autotable (js) : <https://phppot.com/javascript/jspdf-autotable/>
* Recherches sur la fonction typeof (js) : <https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Operators/typeof>
* Recherches sur la fonction switch (js) : <https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Statements/switch>
* <https://www.tutorialspoint.com/index.htm>
* <https://waytolearnx.com/>
* <http://php.adamharvey.name/manual/en/>
* <https://www.php.net/manual/fr/>