

Pré-Ing 2 MI Groupe 6

Rapport du projet Web



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

Sommaire :

• Conception	2
○ Par Nathan.....	2
○ Par Émilien.....	3
○ Par Xavier	4
○ Par Camille.....	5
• Analyse du projet	8
○ Planning.....	8
○ Défauts du site.....	9
• Manuel d'utilisation	10
○ Prérequis à l'utilisation du programme.....	10
○ Utilisation du site Web.....	11
• Manuel technique	13
○ Schéma de l'organisation de nos fichiers.....	13
○ Objectif des fichiers	14
• Bibliographie	15

Conception

Par Nathan :

On a tout d'abord décidé de poser les "fondations" du programme avec un **code HTML** qui, bien que passif, nous permettait de nous représenter l'allure générale du site. Chacun étant chargé d'un module (jeune, référent, consultant et visiteur), j'ai donc tout d'abord été chargé de développer le HTML du module jeune. J'ai donc divisé cette partie en plusieurs fichiers : tout d'abord un pour se connecter au module jeune, un autre pour consulter son compte, un pour s'inscrire et un dernier muni d'un formulaire pour envoyer une demande de référence au référent associé.

Ce HTML a servi dès la deuxième "phase" du développement qui consistait à ajouter les **scripts** nécessaires aux interactions avec le site. Je me suis donc attelé à la partie de la **création de demande** de référence du module jeune qui doit permettre depuis un formulaire d'envoyer une demande de validation de compétences à un référent. Cela s'est donc fait en plusieurs étapes. Premièrement, j'ai récupéré les valeurs entrées dans le formulaire pour les enregistrer dans un fichier. Le fichier est contenu dans un dossier spécifique à chaque utilisateur et avec un nom sous un format très spécifique. Chaque nouvelle référence se voyait attribuer un numéro, le numéro de la dernière référence étant stocké dans un fichier à part et remplacé à chaque création de nouvelle référence. Pour la liste de savoir-faire, je me suis servi de JavaScript pour vérifier le nombre de cases cochées. Un message d'erreur apparaissait après la sélection de 5 cases, et les cases cochées étaient enregistrées dans le même fichier, chaque case étant sur une ligne séparée par des virgules.

Et jusqu'ici tout allait bien, les fichiers se créaient parfaitement, chaque valeur était exploitable. Cependant, le but de cette page était **d'envoyer** la référence, ce qui impliquait d'envoyer un courriel depuis un code PHP. La première étape était de transmettre les variables voulues (ID de l'étudiant et numéro de la référence) dans un lien qui permettrait d'accéder à cette référence enregistrée sur le serveur. J'ai donc utilisé des variables de formulaire GET pour les récupérer depuis le lien. Le lien était ainsi créé vers la future page accessible au référent, et une fonction JavaScript permettait de copier le lien créé après la création de la référence. Il ne restait plus qu'à l'envoyer... Et c'est là que les choses se compliquent.

Quand j'ai commencé mes recherches sur l'envoi de mail depuis du PHP j'ai découvert qu'il y avait une fonction *mail()* native, un jeu d'enfant donc, mais la principale difficulté pour ce genre de tâche n'est pas le code mais toutes ses dépendances et la configuration du serveur d'envoi. J'avais donc besoin d'une bibliothèque me permettant de gérer l'envoi de mail et de me connecter au serveur *smtp* (serveur d'envoi) de l'adresse d'envoi. Internet m'a proposé *PHP Mailer*, la solution la plus courante à ce genre de besoin, la suite était donc simple : je crée une adresse mail quelconque, j'installe *PHPMailer* sur le serveur et je le configure pour qu'il me connecte au serveur *smtp* correspondant au nom de domaine. Rien n'a jamais fonctionné. À chaque essai une erreur différente, je ne compte plus le nombre d'adresses mails nommées « *projetjeunes64@domain* », et les seules sources d'aides possibles dataient de 10 ans sur des forums obscurs.

Mon problème avec les adresses *gmail* étant l'authentification, Monsieur Galouz m'a transmis la méthode pour s'y connecter depuis des applications tierces, piste que j'aurais dû suivre... Cependant on m'a également parlé de *mailgun*, un service de suivi de mails qui proposent également un API pour l'envoi depuis PHP, qui n'a jamais fonctionné mais qui m'a permis de me rendre compte d'une chose : Depuis le début j'utilisais *wamp* pour me servir de serveur de test, et *wamp* est affreux. Ainsi je ne pouvais pas installer *mailgun* puisque pour cela j'avais besoin d'un autre programme d'installation qui avait besoin d'une version récente de PHP, OR *wamp* a une version par défaut de PHP qui date d'il y a 10 ans, et impossible de la remplacer sinon *wamp* ne fonctionne plus (le fichier du nom de « *DO_NOT_DELETE_5.5.12.txt* » l'atteste). C'est ainsi que je me suis tourné vers *xampp*, qui dispose tout simplement nativement d'une application d'envoi de mails, extrêmement simple à configurer et je suis reparti sur une adresse *gmail*, la piste de départ... victoire ...

La **partie référent** s'est ensuite faite très rapidement, le principal étant la simple lecture et affichage d'un fichier de données sur le jeune et d'un autre sur la référence à valider, puis l'envoi d'un formulaire contenant le commentaire du référent et les savoirs-faire validés donc la disposition change dans le fichier pour les différencier. Une des étapes finales a été d'adapter les noms de variables à un fichier global contenant les constantes communes aux différentes pages, et également de s'adapter à l'utilisateur connecté en cours.

Nous nous sommes finalement penchés sur la **soutenance** et j'ai rédigé le manuel d'utilisation (Avec un tutoriel absolument clair sur la configuration du serveur smtp).

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 donné 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 avoir 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 « *& ;* » (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'envoi 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 a 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, etcétera...

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 :

	21/04	21/04	08/05	15/05	22/05	29/05	05/06	12/06
Installation de Wamp/Xamp sur les ordinateurs personnels	Xavier code la partie HTML du référent	Nathan code la partie « <i>Création d'une demande de référence</i> » du module étudiant	Camille code la partie « <i>Envoie des références validées au consultant</i> » du module étudiant	Camille commence la rédaction du rapport	Rédaction du rapport collective			
Création et connexion au répertoire GitHub	Nathan code la partie HTML du module étudiant			Nathan code le module référent	Préparation collective à l'oral			
Découverte de la consigne	Émilien code la partie HTML du visiteur				Camille organise le répertoire GitHub			
Organisation collective (planning, choix de la base de données, etc...)	Camille code la partie HTML du module consultant				Nathan code la partie « <i>envoi de mails</i> »			
					Émilien code le CSS			
					Xavier code la partie « <i>Création de compte</i> », « <i>Connexion</i> » et « <i>Modification du profil</i> » 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 informatique et s'est donc vu attribuer des tâches plus difficiles comme l'envoi 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 descriptions) à d'autres personnes.

Défauts du site :

L'accès aux différents fichiers n'est pas sécurisé. En effet, la communication entre les fichiers se fait par des formulaires de type *GET* ou *POST* sans encodage ; les url peuvent donc être modifiés manuellement pour accès aux fichiers.

Les mots de passes des jeunes ne sont pas encodés. La saisie de celui-ci n'est d'ailleurs pas cachée.

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 caractères.

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

Lorsque qu'un étudiant choisit de créer un PDF avec ses références, le PDF généré contiendra des erreurs s'il le texte de la référence choisit contient des retours à la ligne.

Lors de la création d'un lien (pour envoyer une demande de confirmation de référence ou pour partager une référence), une *alert* se crée.

La mise en page ne s'adapte à la taille de celle-ci.

Manuel d'utilisation :

1) Prérequis à l'utilisation du programme

Configuration du serveur de mails :

Pour le test on utilise le logiciel Xampp qui dispose d'une application d'envois de mails :

1) Installer Xampp (ne rien lancer pour l'instant)

2) Dans le dossier d'installation de Xampp :

- Aller dans xampp\sendmail\sendmail.ini

- Modifier les lignes suivantes : (Pour trouver les lignes utiliser ctrl+F et entrer un bout de la ligne)

 - smtp_server=smtp.gmail.com

 - smtp_port=587

 - smtp_ssl=tls

 - Enlever le « ; » au début des lignes « error_logfile=error.log » et

« debug_logfile=debug.log » pour avoir les messages d'erreur au cas où

- auth_username=leprojetjeunes64@gmail.com (adresse créée pour le projet)

 - auth_password=nenqieyhnrokecf (mot de passe d'application)

3) Également dans le dossier d'installation de xampp

- Aller dans xampp\php\php.ini

- Modifier les lignes suivantes :

 - Ajouter un « ; » au début de la ligne « SMTP=localhost »

 - Ajouter un « ; » au début de la ligne « smtp_port=25 »

 - Ajouter un « ; » au début de la ligne « sendmail_from[...] »

- Trouver le chemin absolu de sendmail.exe

 - Pour ça aller dans xampp\sendmail\sendmail et cliquer droit sur

sendmail.exe => Propriétés et là à droite de « Emplacement » vous avez le chemin absolu du dossier

 - Copier ce chemin

- Trouver dans php.ini la ligne sendmail_path= et ajouter le chemin après le « = » ainsi que \sendmail.exe, on doit se retrouver avec quelque chose qui ressemble à ça :

 - sendmail_path=C:\xampp\sendmail\sendmail.exe

- Bien enlever le « ; » au début de la ligne pour qu'elle soit prise en compte

- Chercher la ligne « extension=php_openssl.dll » et vérifier qu'il n'y a pas de « ; » devant

4) Le code se met dans xampp\htdocs

5) Lancer le serveur Apache de Xampp

6) C'est supposé marcher

2) Utilisation du site Web

Page d'accueil :

Le lien du site envoie vers une page d'accueil présentant brièvement le programme et contenant un bandeau avec différentes sections cliquables pour s'y rendre.

Section jeune :

Lors du clic sur JEUNE du bandeau de la page d'accueil, l'utilisateur est redirigé vers la page d'accueil de l'espace Jeunes qui lui propose de se connecter ou de créer un compte.

-Se connecter :

L'utilisateur est invité à saisir son adresse mail et son mot de passe

-Créer un compte :

L'utilisateur est invité à saisir :

- Nom
- Prénom
- Date de naissance
- E-mail
- Mot de passe

Ces champs doivent être valides pour accéder à la page suivante

Dans les deux cas si les champs remplis sont corrects l'utilisateur est redirigé vers son compte depuis lequel il peut effectuer diverses actions :

-Modifier le profil :

L'utilisateur est dirigé vers une page contenant ses informations dans un tableau prérempli qu'il peut modifier pour changer ses informations personnelles

-Demande de référence :

L'utilisateur est dirigé vers une page permettant d'envoyer une demande de référence à son référent de mission avec plusieurs champs à remplir :

- Description de l'engagement (possiblement sur plusieurs lignes)
- Durée de l'engagement
- Milieu de l'engagement
- Données personnelles du référent
- E-mail du référent
- Liste des savoirs-faire à cocher (au nombre limité de 4, il est impossible d'en cocher plus)

Une fois les champs remplis l'utilisateur peut confirmer pour que sa demande soit envoyée par mail, cependant si cela échoue une option apparaît pour copier le lien de la demande et l'envoyer manuellement au référent.

-Afficher la liste des références :

L'utilisateur est dirigé vers une page contenant la liste de ses références, spécifiant si elles ont été validées ou non par le référent.

-Envoyer les références au consultant :

L'utilisateur est dirigé vers une page permettant de sélectionner les références validées et de les envoyer par mail au consultant choisi, cependant si cela échoue une option apparaît pour copier le lien de la demande et l'envoyer manuellement au référent.

-Imprimer mes références :

L'utilisateur est dirigé vers une page permettant de sélectionner les références validées et de les imprimer, soit depuis une page HTML soit sous un format PDF, le choix du format étant à sélectionner.

Section référent :

Lors du clique sur RÉFÉRENT du bandeau de la page d'accueil, l'utilisateur est redirigé vers la page d'accueil de l'espace Référent avec un court paragraphe explicatif du projet.

Section consultant :

Lors du clique sur CONSULTANT du bandeau de la page d'accueil, l'utilisateur est redirigé vers la page d'accueil de l'espace Consultant avec un court paragraphe explicatif du projet.

Section partenaires :

Lors du clique sur PARTENAIRES du bandeau de la page d'accueil, l'utilisateur est redirigé vers la page d'accueil de l'espace Partenaires avec un court paragraphe explicatif du projet et les logos des partenaires.

Lors de la réception du mail par le référent :

Le lien envoyé au référent contient l'identifiant du jeune et le numéro de la référence à valider, le référent arrive donc sur une page lui affichant les données personnelles du jeune et la référence à valider. Il peut donc sélectionner les savoirs-faire présélectionnés par le jeune pour les valider et ajouter un commentaire. Il est ensuite redirigé vers une page de remerciements.

Lors de la réception du mail par le consultant :

Le lien envoyé au consultant lui permet d'accéder à une page regroupant les références validées du jeune pour qu'il puisse les consulter.

Manuel technique

Schéma de l'organisation de nos fichiers :

Légendes :



: Dossier



: Fichier

écrit par Xavier

écrit par Nathan

écrit par Émilien

écrit par Camille



: « Contient »



: « Peut appeler »
ou « Utilise »



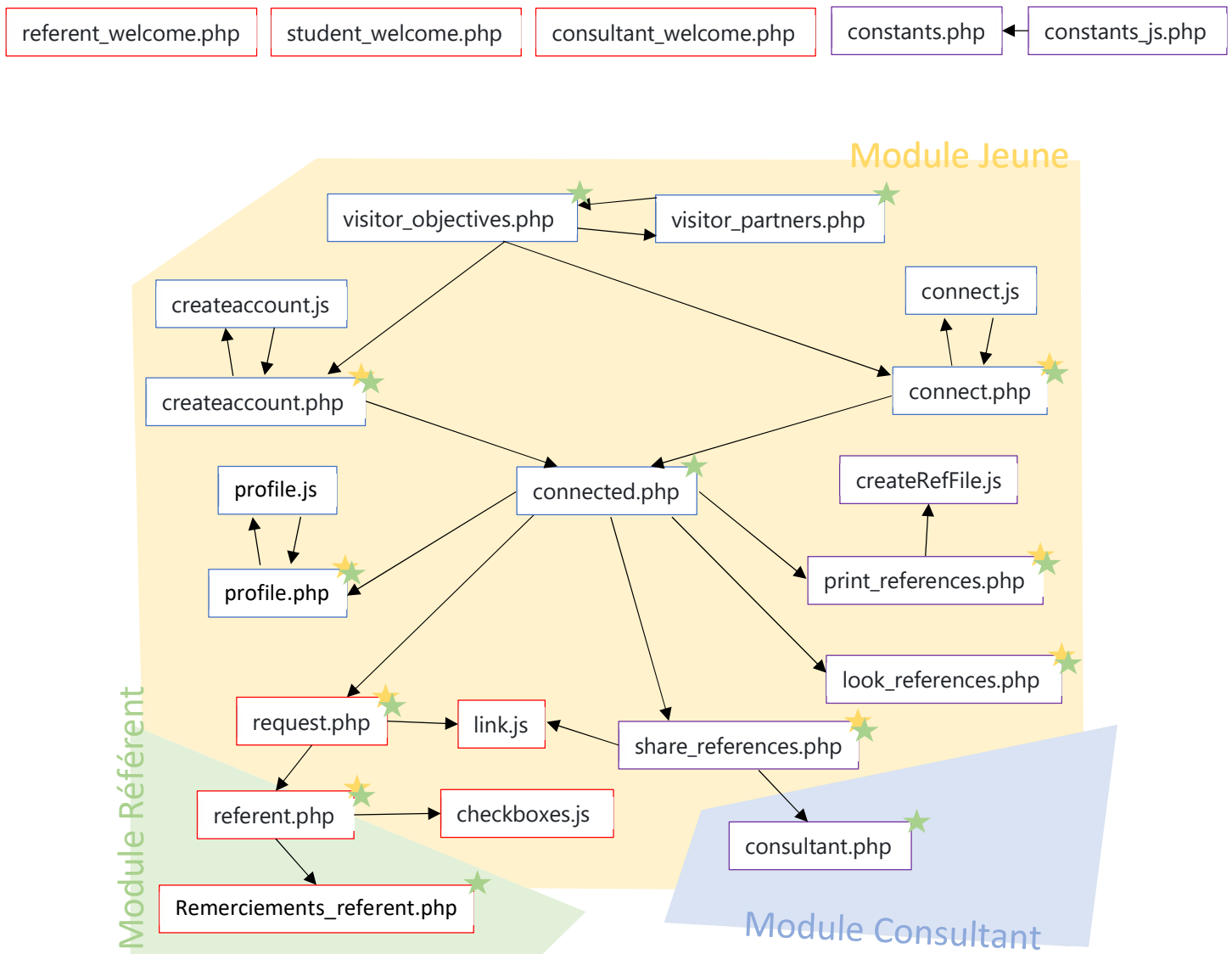
: Utilise le dossier Data

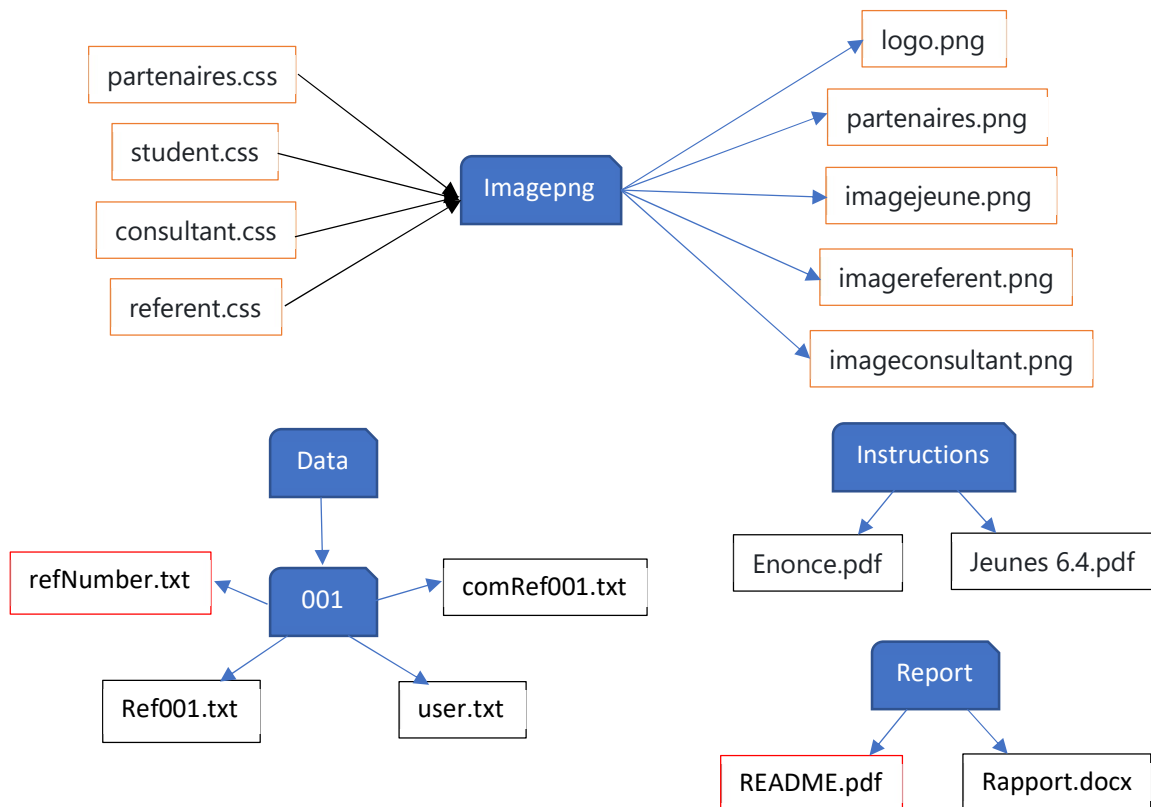


: Appelle le fichier constants.php et
constants_js.php ;

Peut accéder aux pages *_welcome.php* ;

Appelle un fichier .css





Objectif des fichiers :

Les fichiers *constants.php* et *constants.js.js* permettent de générer des variables communes à tous les fichiers. Il contient essentiellement les variables *BANDEAU* (l'entête des pages internet) et le nom des différents fichiers. Le fichier *constants.js.js* inclut des *div* invisibles pour que les programmes en JavaScript puissent également récupérer les variables.

Les fichiers *_welcome.php* permettent de présenter aux utilisateurs le site.

Les fichiers *createaccount.php* et *createaccount.js* permettent de créer un compte jeune à l'aide d'une requête AJAX.

Les fichiers *connect.php* et *connect.js* permettent de se connecter à un compte jeune à l'aide d'une requête AJAX.

Le fichier *connected.php* permet au jeune à l'aide d'une requête AJAX d'afficher les possibilités du jeune une fois connecté (regarder son profil, modifier celui-ci, consulter ses références, les imprimer ou les envoyer).

Les fichiers *profile.php* et *profile.js* permettent à un jeune connecté de modifier son profil jeune à l'aide d'une requête AJAX.

Le fichier *request.php* permet créer une référence et d'envoyer la demande de confirmation au référent. Il utilise le fichier *link.js* qui permet de créer un lien.

Le fichier *referent.php* permet au référent de confirmer une référence. Il utilise le fichier *checkboxes.js*, il permet de s'assurer qu'il y a moins de 5 cases *savoirs-être* cochées.

Le fichier *Remerciements_referent.php* permet de confirmer au référent son action.

Le fichier *share_references.php* permet de partager avec un consultant une référence. Il utilise le fichier *link.js* qui permet de créer un lien.

Le fichier *consultant.php* permet au consultant de consulter des références envoyées.

Le fichier *look_references.php* permet au jeune connecté de regarder ses références.

Les fichiers *print_references.php* et *createRefFile.js* permettent à un jeune connecté de générer ses références sous format HTML ou PDF.

Les fichiers *.css*, permettent la mise en forme de chacun des modules. Ils utilisent les fichiers contenus dans le dossier *Imagepng*.

Bibliographie

Sites fréquemment utilisés :

- Recherches globales :
<https://openai.com/blog/chatgpt>
- Recherches sur différentes balises en HTML :
<https://www.tutorialspoint.com/index.htm>
- Recherches sur de différentes fonctions en JavaScript :
<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/>
- Recherches sur de différentes fonctions en PHP :
<https://www.php.net/manual/fr/>

Sites qui ont permis la résolution de problèmes spécifiques :

- Vidéo qui a permis l'envoi de mails :
<https://youtu.be/rLFbfcTCBM8>
- 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 (en JavaScript) :
<https://stackoverflow.com/questions/16858954/how-to-properly-use-jspdf-library>
- Recherches sur les bibliothèques jsPDF Autotable (en JavaScript) :
<https://phppot.com/javascript/jspdf-autotable/>