

Projet Informatique CY-TECH 2022

Dev Weeb



CY Tech

Table des matières :

1. Intro
2. Choix techniques
3. Répartition des tâches au sein du groupe
4. Description du système de fichiers choisi

Intro :

Pour ce projet il nous a été demandé de réaliser un site permettant de répartir les futurs ING3 dans leurs options en fonction de leur moyenne et de leurs vœux. C'est en quelques sortes le Parcoursup de l'ING2. Ce site devra aussi permettre une gestion de différents types de comptes ainsi que l'utilisation d'une messagerie intégrée. Il sera donc possible de modifier son profil, remonter une erreur, communiquer avec les autres utilisateurs, et bien plus encore.

Bienvenue, dans OwOption – CY-Tech

Choix techniques :

Messagerie :

Tout d'abord nous allons parler de la messagerie. Nous avons utilisé de l'AJAX pour l'implémenter, afin que les messages se mettent à jour sans avoir à recharger la page.

Le problème principal sur lequel nous nous sommes heurtés est la modification des fichiers, en effet nous devons pouvoir ajouter et surtout supprimer des blocs d'informations facilement et c'est donc pourquoi nous avons choisis un système de fichier en JSON plutôt que CSV. En PHP les fonctions `decode` et `encode` nous permet de modifier toute la donnée de manière très rapide comme le montre les images si dessous.

```
<?php
    echo $_POST['id'];
    $id = $_POST['id'];

    $data = file_get_contents("./conv/".$_POST["adress"]);
    $json = json_decode($data);

    unset($json[$id]);

    $json = json_encode(array_values($json));
    file_put_contents('./conv/'.$_POST["adress"], $json);
?>
```

Un simple `unset` afin de supprimer un message en 7 lignes de codes

```
<?php

    $data = file_get_contents('./conv/'.$_POST["adresse"]);
    $json = json_decode($data);

    $array = array(
        'emetteur' => $_POST['emetteur'],
        'message' => $_POST['message'],
        'state' => "visible"
    );

    $json[] = $array;

    $json = json_encode($json, JSON_PRETTY_PRINT);

    file_put_contents('./conv/'.$_POST["adresse"], $json);

    echo $_POST['message'];
?>
```

Nous avons ensuite choisis une arborescence de discussion en 2 parties :

Un fichier conv.json que nous pouvons facilement traverser et qui contient toutes les informations sur les discussions en cours et le nom du fichier contenant la dîtes conversation

Un dossier conv qui contient un fichier par discussion lancé entre 2 utilisateurs.

Ce choix a était fait afin de favoriser une meilleur rapidité pour charger les conversations, nous avons qu'un seul fichier réduit à traverser et analyser au complet afin de connaître si la conversation existe et son nom. Nous n'avions ensuite plus qu'à ouvrir le fichier associé et affiché les messages. Cette méthode était bien plus rapide que de paquet tous les échanges de message dans 1 JSON qui nous devons traverser et analyser à chaque itération.

```
▼ conv
{} Denis-AdrienAlves@cy-tech.frMarcRibeiro@cy-tech.fr.json
{} Denis-AdrienAlves@cy-tech.frMartheSeguin@cy-tech.fr.json
{} Denis-AdrienAlves@cy-tech.frSusanSauvage@cy-tech.fr.json
{} conv.json
```

Modules :

A contrario, la modification de compte se fait par PHP, la page se recharge donc, même si cela est fait automatiquement. Cela s'explique par le fait que les membres du groupe s'étant chargé de ces différentes parties n'ont pas les mêmes préférences de langage. Certains ont préféré le PHP ou JavaScript, et inversement.

Pour les logs et les erreurs, nous avons choisi d'utiliser un fichier .txt (sans extension) car nous n'avions pas besoin de le transformer en tableau par la suite.

Concernant les fichiers Listes[].csv, nous avons décidé d'utiliser un login et un mot de passe prédéfini pour le responsable admission ainsi que l'admin. Cependant les logins et les mots de passe des étudiants sont, eux, générés automatiquement. Le mot de passe est généré aléatoirement grâce à la fonction « uniqid », une fonction générant une chaîne de caractères aléatoires en se basant sur le fuseau horaire.

```
$mdpRandom = uniqid(); /*On génère un mdp aléatoire */
```

Pour le login, nous avons choisi d'utiliser l'adresse CY-Tech, pour uniformiser les logins et faciliter leur utilisation dans le cadre de la messagerie notamment. Ainsi les logins seront de la forme : PrenomNom@cy-tech.fr
Finalement, le profil d'un étudiant ressemblera à ceci :

Marc;Ribeiro;;;MarcRibeiro@cy-tech.fr;628fc06fd7f10

Plusieurs champs étant laissés vides lors de la génération automatique car ils sont inconnus de l'administrateur (ex : photo de profil, adresse, téléphone, etc). Il reviendra donc à l'étudiant de les remplir lors de sa première connexion.

Concernant la photo de profil, étant donné qu'il était trop complexe de permettre à l'utilisateur d'en importer une depuis sa machine, nous avons choisi de lui donner le choix entre une dizaine d'images différentes, préenregistrées dans un dossier du projet.

CSS :

Concernant le CSS du site, nous avons décidé d'avoir une approche mimi du projet. Si attirer les cœurs était le but premier, nous pouvons à présent garantir avoir appris énormes de chose en CSS.

D'abord, nous avons envie de faire une premier page "accueil.php" globale pour en mettre plein la vue en guise d'échauffement pour le reste du projet. Nous avons donc utilisé du JS mêlé avec du CSS (PHOTO 1) et des keyframes CSS (PHOTO 2) pour créer des petites animations sympa. Cela nous a pris bien plus de temps qu'il ne l'aurait fallu mais nous étions encore naïfs concernant la taille du contenu à traiter par la suite...

```
<script type="text/javascript">
function opentab() {
    document.getElementById("arrow").style = "transform: rotate(180deg);margin-left: 15em;";
    document.getElementById("selectbox").style = "width: 18em; transform: scale(1.2);";
    document.getElementById('arrow').setAttribute( "onClick", "closetab();" );
    document.getElementById('log').style = "opacity: 100%;";
    document.getElementById('reg').style = "opacity: 100%;";
}

function closetab() {
    document.getElementById("arrow").style = "transform: rotate(0deg);margin-left: 0em;";
    document.getElementById("selectbox").style = "width: 4em;";
    document.getElementById('arrow').setAttribute( "onClick", "opentab();" );
    document.getElementById('log').style = "opacity: 0%;";
    document.getElementById('reg').style = "opacity: 0%;";
}
```

Photo 1

```

.content{
  font-size: 150%;
  margin: 0;
  position: absolute;
  top: 50%;
  left: 30%;
  margin-left: 2em;
  -ms-transform: translate(-50%, -50%);|
  transform: translate(-50%, -50%);
  animation: appear 5s;
}

@keyframes appear {
  from{
    left: 55%;
    opacity: 0%;
  }
  to{
    opacity: 100%;
  }
}

#arrow{

```

Photo 2

Après, nous avons besoin de créer un menu permettant de naviguer entre les différentes fonctionnalités du site. Nous avons donc créé "haut.php" qui allait être include tout en haut de chaque page.

```

1 <?php include "haut.php" ?>

```

Tout le contenu des pages est placé dans une div "content" permettant d'uniformiser l'apparence des pages. Une fois le squelette terminé, nous avons sectionné le travail en 3 parties :

- Partie connexion : l'apparence de la page de connexion classique par excellence avec un petit système de notification en utilisant les redirections "header(Location ...)" avec la méthode "GET". Particulièrement utile pour ramener les plus pressés ou les plus filous à contourner le bon fonctionnement du site.
- Partie Profil : Dans cette partie, nous avons décidé de faire un seul et unique CSS pour les 3 types de module. Comme "admission" et "élève" se ressemblent et que le

module admin ne comporte pas énormément d'éléments à styliser, tout faire en un unique CSS était faisable. Nous aurions aimé passer plus de temps sur le fonctionnement et le design des boutons pour charger les fichiers ".csv" mais après quelques recherches, nous avons vite compris que cela était un peu trop long et compliqué à mettre en œuvre.

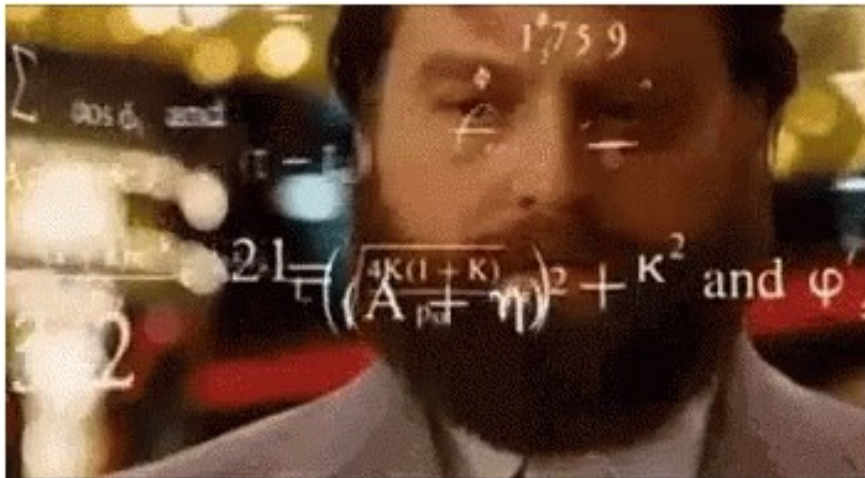
- Partie Messagerie : De LOIN la partie la plus ardue que nous avons eu à traiter. Nécessité de manier le JS/DOM/CSS très rigoureusement pour faire en sorte que la boîte de la messagerie ne se casse à la moindre interaction. Que ce soit au chargement de la page, à l'affichage d'une discussion, à l'envoi d'un message ou même à la création d'une nouvelle conversation, tout était prétexte à remettre en question le CSS construit en amont !

Pour finir, bien que nous nous soyons seulement focalisé sur l'apparence du site, nous avons dû à de nombreuses occasions, revoir certaine partie du code des autres pour que ça puisse "coller". Typiquement rajouter des "class", des "id" ou même changer l'arborescence des fichiers.

C'est d'ailleurs ce qui nous chagrine le plus dans ce que nous avons fait. Conséquences d'une mise en commun trop tardive, nous nous sommes rendu compte que nous n'avions pas organisé nos ".php" comme les autres. Il a fallu donc tout mettre "au même niveau" ce qui rend l'archive relativement brouillonne... Mais à part ça, nous sommes satisfaits du travail rendu !

Répartition des tâches au sein du groupe :

- Nicolas : Algorithme du Mariage Stable + Messagerie
- Anthony : Messagerie
- Léo-Paul : CSS Master + Uniformisation et mise en commun des parties
- Gautier : Module Responsable Admission
- Titouan : Module Etudiants + Module Responsable Admission + Module Administrateur



Description du système de fichiers choisi:

Par soucis de simplicité, nous avons décidé de ramener la plupart des fichiers à la racine lors de la mise en commun, afin de pallier à tous les problèmes de chemin (que nous n'avions pas le temps de résoudre autrement). En effet, nous avons chacun créé une arborescence de notre côté et l'utilisation de certains « include » (appelés dans différentes pages à différents niveaux) n'était pas compatible avec chacune de nos arborescence. Il nous aurait fallu revoir le tout pour corriger chacun soucis de chemin. Nous avons décidé d'aller au plus simple et d'utiliser notre temps à perfectionner le site, jugeant que l'esthétique de l'arborescence n'était pas un point majeur.

