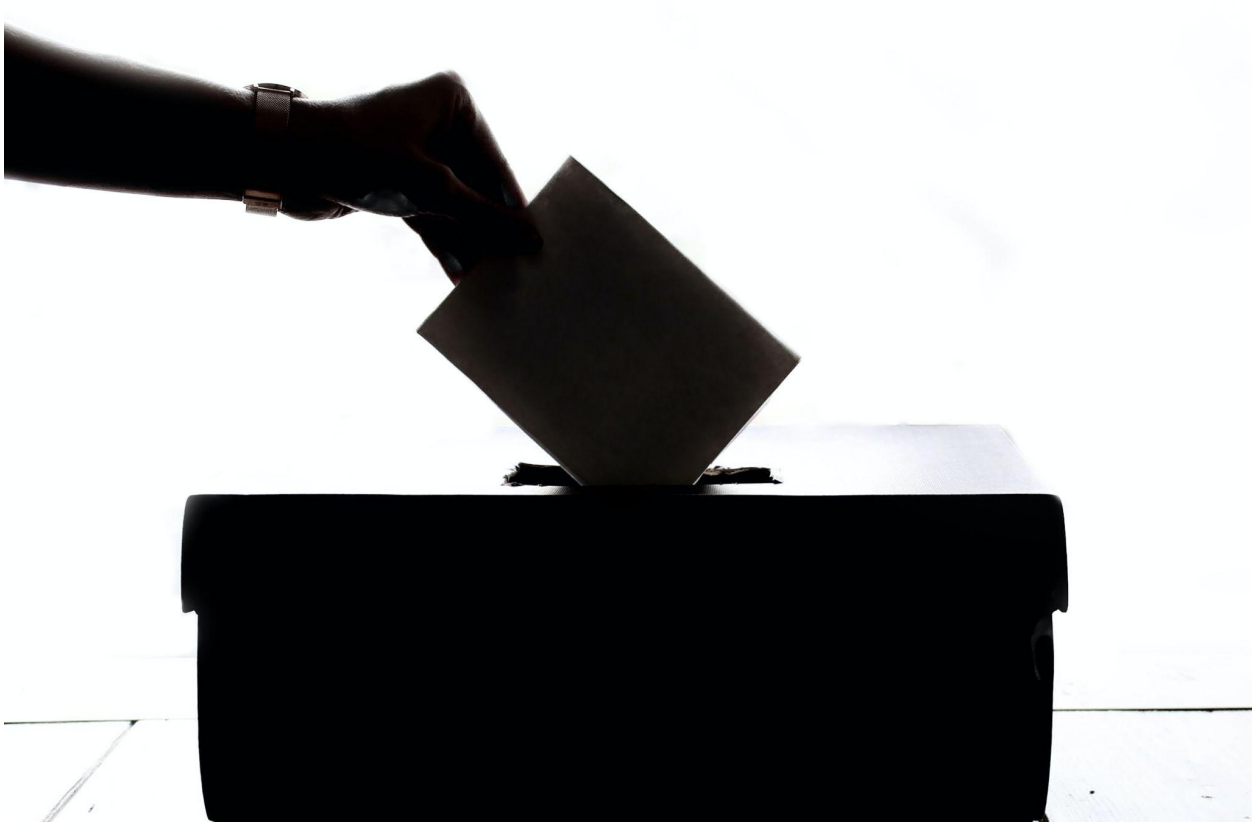
Système de vote en ligne



HIRLÈS Solène

2021-2022

WEB – APP3 - INFO

INTRODUCTION

Depuis 2020, nous vivons dans un contexte difficile de pandémie. Nous avons donc changé notre méthode de travail en optant pour du télétravail. Néanmoins, nous essayons quand même de maintenir des réunions dans lesquelles nous prenons des décisions pour l'avenir de l'entreprise. Pour cela, la visioconférence est l'outil parfait. Le seul problème avec cet outil est qu'il ne permet pas d'organiser un vote tout en gardant les réponses anonymes. Pour pallier ce problème, je vous propose une application web permettant de créer des sondages et de répondre à ceux-ci.

Bien que cette application soit utile dans ce contexte, quand celui-ci se terminera alors elle n'aura plus d'utilité. C'est pourquoi la plateforme que je vous propose permet également de faire des sondages publics où tout le monde peut participer. Cette fonctionnalité a pour but d'aider les personnes qui en ont besoin pour leurs recherches notamment des étudiants dans le cadre de leurs études ou encore des entreprises pour étudier les besoins de leurs clients. De plus, pour être utilisable dans n'importe quel domaine, les utilisateurs peuvent éditer les questions et les choix de réponses.

CAHIER DES CHARGES

1- Page d'accueil

L'utilisateur doit pouvoir s'inscrire, se connecter et se déconnecter. Un compte est créé avec une adresse mail, un mot de passe et un pseudo. **L'adresse mail doit être unique et valide**. Le pseudo doit être **unique**, doit contenir plus de 3 caractères (**4 minimum**) et commencé par une majuscule.
De plus, le caractère **'@' est interdit** dans les pseudos.
Pseudos et email sont **non modifiables** après création.

L'adresse email servira pour recevoir des mails tandis que le pseudo apparaîtra comme auteur des sondages.

2- Page de création de sondage

Une fois connecté, on doit pouvoir créer un sondage.
Il comprendra les items suivants :

- Un titre
- La portée du sondage (public ou privé)
- L'utilisateur devra rentrer **au moins une question et au maximum 10**.
- Les choix de réponses seront par défaut "oui", "non", "Abstention". Néanmoins, l'utilisateur pourra ajouter ou supprimer des choix de réponses. Il faut **au minimum 2 choix de réponses par question** pour que le sondage soit valide donc créé. Il pourra saisir 5 choix de réponses au maximum
- Il faut définir une date de clôture, **au maximum 1 an après la date de création**. (Après cette date les réponses ne seront plus prises en compte)

Si le sondage est publique, on aura en plus les items suivants:

- un thème (**en 1 mot** pour permettre la recherche du sondage).

Si le sondage est privé, ce seront ceux-là:

- une liste d'électeurs (avec des pseudos ou des emails, **un compte ne peut apparaître qu'une fois dans la liste**).

Quand le sondage est validé, un **identifiant unique** lui est attribué. On enregistre en plus des items du sondage, l**a date de création du sondage, le pseudo et l'email** du créateur.

Une fois validé, pour les sondages privés, **un mail avec le lien et l'identifiant du sondage est envoyé à la liste d'électeurs.**

3- Page de recherches de sondages

L'utilisateur pourra **chercher un sondage** grâce à un formulaire. La recherche peut se faire grâce à :

- L'identifiant
- Le thème (**pour les sondages publiques**)
- La date de clôture (**pour les sondages publiques)**.
- Titre (**publiques**)
- pseudos du créateur (**publiques**)

4- Page de vote

L'utilisateur pourra voter à tous les sondages publiques et aux sondages privés auxquels il a accès.
Il ne peut voter qu'une fois par sondage.
Une réponse à chaque question est nécessaire pour valider le vote.
Un vote non complet n'est pas pris en compte.
Le date de clôture ne doit pas être passée.
Lors du vote, on enregistre **la date du vote** ainsi que les réponses et que le votant a déjà voté.

5- Page du compte de l'utilisateur

Sur son profil, l'utilisateur pourra voir ses sondages créés et les gérer.
Ce sera un **affichage sous forme de tableau** avec les colonnes suivantes: identifiant, titre, visibilité (publique ou privé), date de clôture. Il y aura également une colonne pour sélectionner les lignes.

Il pourra réaliser les actions suivantes:

- Modifier la visibilité des résultats
- Modifier la date de fin **(max 1 an après la date de création)**
- Afficher le nombre de réponses/ taux de participation
- Fermer le sondage et donc afficher les résultats (**bouton avec message d'alerte + modification de la date de clôture)**
- Supprimer le sondage (**bouton**)

Il pourra aussi voir les sondages auxquels il a voté qui seront affichés **sous forme de tableau** avec *identifiant, titre, pseudo du créateur, date de réponse*. La colonne suivante affichera **soit la date de clôture si le sondage est encore ouvert soit qu'il est fermé**. La dernière colonne proposera un bouton "voir les résultats" qui sera **cliquable uniquement si le sondage est fermé.**

6- Page de résultats

Les résultats seront affichés sous forme de graphiques fait avec **Plotly.** On disposera d'un tableau contenant **des informations importantes** sur le sondage :

- le nombre de votes
- le nombre de votes attendus (pour un **sondage privé**)
- la date de création
- la date de clôture
- la durée du sondage

On aura également **un graphique avec la date des votes.**
Chaque graphique pourra être **copié automatiquement en cliquant sur un bouton**.

7- Bonus, si il reste du temps

La page de résultat peut être **téléchargeable en format pdf.**
L'utilisateur pourra choisir le **type de graphique** qu'il souhaite.
Les graphiques pourront **être déplaçables** pour une meilleure étude.
Les sondages s'**autodétruisent** 1 an après leur date de clôture.
Le créateur pourra choisir **le nombre de réponses que le votant peut sélectionner au maximum** pour chaque question du sondage.
Le créateur du sondage pourra **choisir qui peut voir les résultats**.

8- Technologies envisagées

On utilisera les langages : **HTML, CSS, PHP, JavaScript, SQL**.
On se servira de la librairie **Plotly** Js pour les graphiques et peut-être **Tabulator** pour les tableaux.
En framework, on utilisera soit **W3.CSS** soit **Bootstrap**.

PLANIFICATION

Prévisionnel :

- 6 au 12 Décembre 2021 :
  - Création de la base de donnée
  - Début des fonctions de connexion et d'inscription

- 13 au 19 Décembre 2021 :
  - Finir les fonctions de connexion et d'inscription
  - Gestion des sessions d'utilisateur

- 20 au 26 Décembre 2021 :
  - Création de la page d'accueil
  - Tests de la page
  - Faire une maquette de la page de création de sondage

- 27 Décembre 2021 au 2 Janvier 2022 :
  - Réalisation du HTML de la page de création de sondage
  - Réalisation des fonctions spécifiques au sondage privé/publique

- 3 au 9 Janvier 2022 :
  - Réalisation des fonctions d'ajout des réponses et des électeurs
  - Réalisation des fonctions de suppressions des réponses et des électeurs

- 10 au 16 Janvier 2022 :
    - Envoi du formulaire de création de sondage
    - Remplissage la base de données pour les sondages publiques

- 17 au 23 Janvier 2022 :
    - Vérifier l'existence d'un électeur quand il est ajouté au sondage
    - Vérifier que l'électeur est une seule fois dans la liste des électeurs

- 24 au 30 Janvier 2022 :
    - Remplir la base de données lors de sondages privés
    - Maquette de la page de vote
    - Code HTML de la page de vote

- 31 Janvier au 6 Février 2022 :
    - Remplissage de la page de vote avec la base de données
    - Sécuriser l'accès à la page de vote

- 7 au 13 Février 2022 :
    - Ajout du vote dans la base de donnée
    - Tests des pages de création de sondage et de votes

- 14 au 20 Février 2022 :
    - Maquette de la page de recherche de sondages
    - Affichage de le tableau de tous les sondages

- 21 au 27 Février 2022 :
    - Ajout des filtres sur le tableau des sondages
    - Création de la page profil avec les tableaux d'historique des votes et des créations de sondages

- 28 Février au 6 Mars 2022 :
    - Ajout des fonctions de la page profil
    - Début de la page de résultats

- 7 Mars au 13 Mars 2022 :
    - Fin de la page de résultats
    - Réalisation du style graphique des pages

Réalité

- 6 au 12 Décembre 2021 :
    - Début de la base de données
    - Fonction inscription

- 13 au 19 Décembre 2021 :
    - Fin de la base de données
    - Fonction connexion
    - Préparation des sessions

- 20 Décembre 2021 au 2 Janvier 2022 : (2 semaines)
    - Recherche et essaie de résolution du problème de passage de variables entre PHP et JavaScript

- 3 au 16 Janvier 2022 : (2 semaines)
    - Découvertes et tests des frameworks de W3.css et Bootstrap

- 17 au 23 Janvier 2022 :
    - Préparation du HTML de la page de création de sondages

- 24 au 30 Janvier 2022 :
    - Fonction d'ajout de réponses aux sondages
    - Gestion de l'envoi des différentes réponses possible aux sondages

- 31 Janvier au 6 Février 2022 :
    - Amélioration de la base de données
    - Fonction de suppression des réponses

- 7 au 13 Février 2022 :
    - Fonction faisant passer le formulaire de sondage public à sondage privé
    - Recherche afin de trouver comment vérifier l'existence d'un électeur

- 14 au 27 Février 2022 :
    - Fonction vérifiant l'existence d'un électeur
    - Remplissage de la base de données pour créer un sondage

- 28 Février au 6 Mars 2022 :
    - Début de la page de recherche de sondage
    - Recherche sur comment passer un tableau en PHP à une variable JavaScript
    - Réflexion sur comment passer à la page de vote avec l'identifiant du sondage de manière sécurisée.

- 7 au 13 Mars 2022 :
    - Fin de la page de recherche de sondage
    - Page du profil
    - Page des résultats
    - Page de vote

Retour sur la planification

Le programme prévu était trop dense à tenir en parallèle des cours et de l'entreprise. J'aurais dû diminuer le nombre de fonctionnalités attendues en me concentrant sur celles qui sont vraiment essentielles. De plus, je n'avais pas tenu compte des problèmes que je rencontrerai ni de la difficulté qu'il me poserait. J'ai perdu énormément de temps sur les problèmes de passages de variables entre les différents langages. Cela n'aurait pas pu être évité. Je détaillerai cela plus profondément dans la suite du rapport.

Un autre point important est le fait de devoir utiliser un framework. Je n'en avais jamais utilisé et j'ai eu des difficultés à comprendre leur fonctionnement et leur utilité notamment à cause de la documentation. En effet, j'ai toujours été obligée de coder le CSS à la main et je préférais cela car j'avais un plus grand contrôle du style et des éléments. C'est également quelque chose que je détaillerai plus tard.

DÉVELOPPEMENT

Technologies utilisés

Pour réaliser ce projet, j'ai utilisé HTML et CSS car ce sont les langages de base du web. Le HTML est le langage qui permet de structurer sa page web. Il est indispensable. A ce langage, j'ai ajouté le CSS dont le but est d'apporter un style graphique à la page web afin de la rendre plus attrayante.

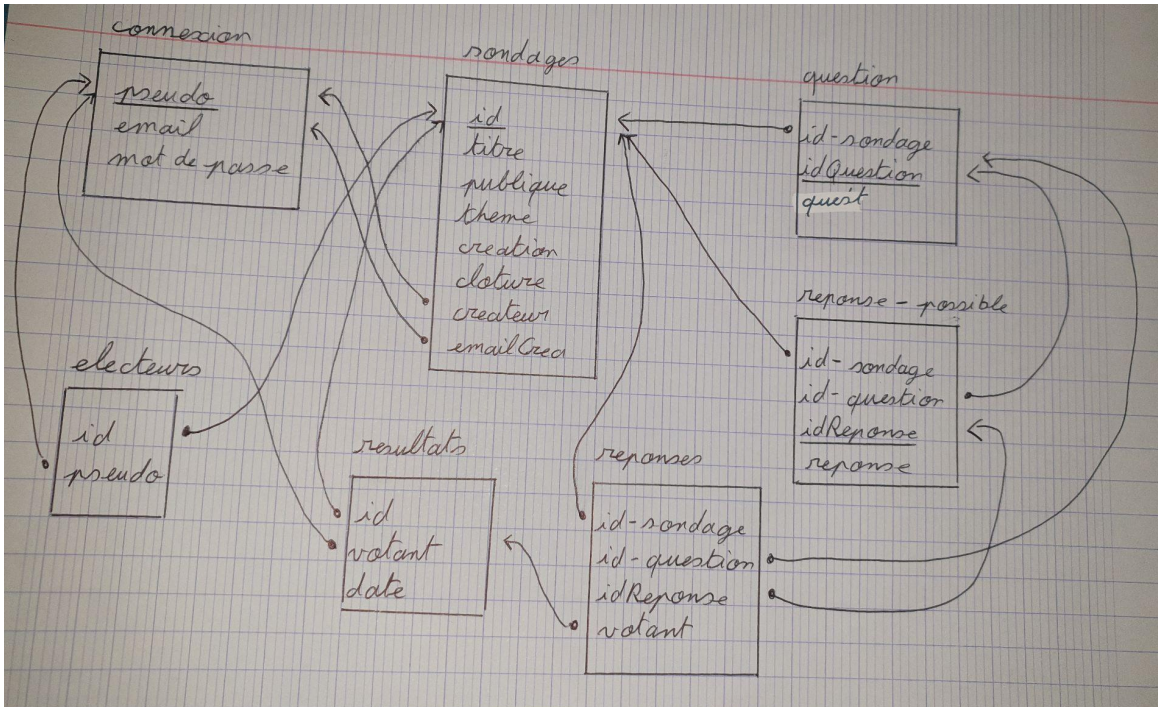
J'ai ensuite ajouté du JavaScript pour rendre la page plus dynamique sans avoir besoin de la recharger. En effet, selon les actions de l'utilisateur, des éléments de la page web peuvent bouger grâce à ce langage. Par exemple, il permet de faire apparaître la fenêtre de connexion lorsque l'utilisateur clique sur le bouton de connexion.

Pour stocker les données, j'ai opté pour une base de données vu la quantité d'informations qui doivent être gardées pour un sondage. J'ai donc dû utiliser du SQL. Afin d'avoir accès à celle-ci et de faire du traitement de données, j'avais besoin d'un vrai langage de programmation, le PHP. De plus, ce langage s'est révélé indispensable pour gérer les sessions d'utilisateur et pour passer des variables entre les pages.

Au niveau des librairies, j'ai utilisé Plotly car c'est la meilleure des librairies existantes pour faire des graphiques en web. On va utiliser des graphiques pour afficher les résultats de manière plus visuelle. Comme je veux afficher une partie de la base de données sous forme de tableau, je vais avoir besoin d'une librairie créant des tableaux simplement mais permettant des fonctionnalités de tris, de filtres et de customisation des données et types de données. J'ai donc pris Tabulator.

Pour finir, il fallait choisir un framework. J'ai hésité entre Bootstrap et W3.CSS. J'ai finalement opté pour W3.CSS car il me suffit amplement pour le niveau de style dont j'ai besoin. De plus, je le trouve plus clair à utiliser notamment grâce à sa documentation qui est très explicite avec beaucoup d'exemples et de tests en ligne. J'ai également testé Bootstrap mais j'ai vite été perdue sur le site et dans le code à cause de la documentation qui n'est pas assez explicite pour moi. Je n'ai donc pas réussi à m'approprier ce framework alors qu'il paraît qu'il est très bon au niveau du web responsive.

Schéma de la base de données



La base de données est composée de 7 tables différentes. Sur le schéma, une table est représentée par un rectangle et son nom se situe au-dessus. Les attributs soulignés sont les clés primaires de la table. Les clés étrangères ont été représentées par une flèches qui part d'elles jusqu'à l'objet auquel elles font références. Je vais maintenant détailler les tables.

La table connexion permet de stocker le compte des utilisateurs de la plateforme. Afin que deux personnes ne puissent pas avoir le même email et/ou le même pseudo, j'ai appliqué une clé primaire sur pseudo qui nous servira à identifier les utilisateurs et j'ai ajouté les contraintes d'unicité et de non-nul sur email. L'attribut "mot de passe" stocke les mots de passe. Néanmoins, ils ne sont pas cryptés. C'est une des améliorations qui peut être envisagée.

La table “sondages” a pour but de stocker toutes les informations d’un sondage. Elle stocke son identifiant qui est unique et permet de l’identifier, son titre et son thème. Elle a également l’attribut “publique” qui est un booléen indiquant si le sondage est public, par défaut c’est le cas. “creation” est la date de création du sondage qui par défaut vaut le jour où une nouvelle est insérée dans la table. Dans “createur” et “emailCrea”, on stocke les informations du compte qui a créé le sondage. Et pour finir, on met, dans “cloture”, la date à laquelle le sondage se termine.

Une des tables liées à sondages est “question”, elle a pour but de stocker une question du sondage et de lui donner un identifiant afin de la représenter. L’autre table liées à “sondages” est “reponse-possible” qui stocke les reponses que pourront sélectionner les votant à une question d’un sondage. Ces deux éléments sont identifiés grâce à leur identifiant. On en donne un également à la réponse afin de pouvoir la représenter plus facilement.

Pour les sondages privés, on a la table "electeurs" qui stocke les pseudos qui ont le droit de voter aux sondages auxquels ils sont associés.

Pour enregistrer les votes, on a la table “resultats” qui stocke le pseudo de la personne qui vote ainsi que la date et l’identifiant du sondage. On enregistre ses réponses aux questions dans “reponses” en reportant le numéro du sondage, de la question et de la réponse choisie.

Tâches réalisées

1- Page d'accueil

Pour la page d'accueil, j'ai choisi de mettre les formulaire d'inscription et de connexion dans un pop-up afin d'alléger la page. En effet, c'est la première page du site sur laquelle on tombe. Elle doit donc être agréable et donner envie de continuer sur le site. Or tomber tout de suite sur deux formulaires n'est pas très plaisant. De plus, j'ai connecté les deux pop-up entre eux. J'ai fait cela afin d'éviter à un utilisateur qui a cliqué sans faire exprès sur connexion mais qui ne possède pas de compte (ou inversement), de devoir fermer le pop-up et d'ouvrir le deuxième. La fonctionnalité que j'ai ajouté permet de gagner du temps et donne envie de poursuivre.

De plus, c'est dans cette partie de connexion qu'est survenu mon premier problème. En effet, lorsque l'utilisateur est connecté, je n'ai plus besoin d'afficher les boutons de connexion et d'inscription mais j'ai besoin d'afficher le bouton de déconnexion. Or, le traitement de la connexion et de l'inscription s'effectue en PHP. Si la connexion est bien effectuée, on passe à l'objet session un champ indiquant que l'utilisateur est connecté et ce champ vaut vrai. Or la gestion de l'affichage de la page HTML s'effectue en JavaScript car les boutons dont nous avons parlé contiennent des fonctions JavaScript(JS) permettant d'ouvrir les pop-up. Il fallait donc passer une variable PHP à une variable JS. J'ai effectué beaucoup de recherches dont certaines qui demandaient l'utilisation du JQuery. J'ai essayé d'importer cette librairie mais sans succès. D'autres demandaient de passer par une requête HTTP sauf qu'on ne peut pas envoyer de requête d'un PHP vers un HTML car on n'a pas moyen de récupérer les données. Après plusieurs jours de recherches, il s'est révélé que la solution était relativement simple car elle consistait à afficher la variable PHP dans une variable JS mais ça ne fonctionne pas pour tout type de variable. Nous y reviendrons dans la suite.

Plus tard, lors du développement de la page de recherche, j'ai repensé à cette page d'accueil. Hormis les boutons pour naviguer de page en page ou pour se connecter et s'inscrire, la page était très vide. J'ai donc pris la décision de compléter l'accueil avec la page de recherche. De plus, cela donne tout de suite un aperçu du site aux nouveaux utilisateurs quand ils arrivent. Néanmoins, j'ai dû adapter cette partie. En effet, il y avait une colonne qui permet de voter au sondage. Or, comme n'importe qui a accès à la page d'accueil, j'ai dû enlever cette colonne. Mais je ne l'ai pas enlever complètement sinon, il n'y a aucun moyen pour voter à un sondage. J'ai donc distinguer deux cas: la cas où l'utilisateur est connecté alors il a accès à ce bouton de vote sinon il ne le voit pas.

Sur cette partie de recherche, j'ai dû réfléchir aux données des sondages qui devaient apparaître. J'ai donc mis l'identifiant du sondage, le titre, le thème (même si il peut être vide car il n'a rien de confidentiel), le créateur, la date de clôture. En plus de ces informations, j'ai rajouté la visibilité afin de permettre de savoir à quel sondage ils peuvent voter librement. Un autre problème s'est alors posé : que fait-on des sondages clôturés? Or, un sondage clôturé est un sondage auquel les utilisateurs ne peuvent pas voter donc nous n'avons aucune utilité à les afficher. Ce bouton lance une API PHP qui récupère l'identifiant du sondage et le met dans la variable session. Une fois que l'API a finit de s'exécuter, on redirige l'utilisateur vers la page de vote.

Pour les fonctionnalités de recherches, l'utilisateur doit pouvoir chercher un sondage selon n'importe quelles colonnes. J'ai donc créé une liste déroulante qui permet de choisir selon quelle colonne, on veut faire la recherche. Il ne reste plus qu'à l'utilisateur à écrire le début de ce qu'il cherche.

Dans la partie de recherche, j'ai eu un autre problème de passage de variable de PHP à JS. En effet, j'ai dû faire passer les tableaux contenant les résultats des requêtes SQL pour pouvoir les afficher. J'ai essayé la même méthode que précédemment mais cela ne fonctionne pas pour des objets (car les tableaux sont des objets). Je me suis souvenu que PHP et JS ont tous les deux des fonctions pour décoder et encoder du JSON. J'ai alors opté pour encoder mon tableau en JSON. Je n'ai pas eu besoin de le décoder car ce tableau a été donné à Tabulator comme données pour le tableau. Il n'accepte que des données sous format JSON.

2- Création de sondage

Pour cette page, j'ai juste créer les champs HTML dont j'ai besoin pour remplir ma base de données. Néanmoins, quand l'utilisateur choisit de faire un sondage privé (rappelons qu'un sondage est public par défaut), il faut faire disparaître le thème du sondage car cela n'a pas réellement d'utilité car seules certaines personnes pourront voter aux sondages. Elles seront donc au courant de quoi il parle. De plus, il faut apparaître le champ des électeurs qui permet à l'utilisateur de choisir qui pourra voter.

Les champs réponses et électeurs sont en réalité des tableaux auxquels on peut ajouter et/ou supprimer des lignes. Néanmoins, on ne peut pas passer des tableaux dans un formulaire. J'ai donc dû réfléchir à une solution. Le format le plus simple pour transmettre une donnée à un formulaire est une chaîne de caractères. Il ne restait plus qu'à trouver comment séparer les différents éléments dans cette chaîne. Pour cela, j'ai regardé sur mon clavier quel est le caractère le moins utilisé afin d'éviter qu'il se retrouve dans une réponse et donc qu'il ne soit pas pris en compte lors du traitement. J'ai fini par choisir le caractère '¤'. Après l'envoi du PHP, il ne restait plus qu'à couper la chaîne à chaque fois qu'on croisait ce caractère pour obtenir les informations.

J'ai rencontré une autre difficulté dans cette partie. Il s'agit de vérifier si l'électeur qu'on ajoute existe bien. L'ajout de l'électeur est géré en JS. Or pour savoir si l'électeur existe, j'ai besoin d'avoir accès à la base de données. Pour cela, j'ai créé une petite API en PHP qui récupère l'utilisateur qui doit être ajouté et vérifié s'il est dans la base. Dans un premier temps, l'API renvoie vrai ou faux. Pour envoyer la requête à l'API, j'ai utilisé une requête HTTP. C'est à ce moment que j'ai eu une autre difficulté. En effet, quand on envoie une requête HTTP, celle-ci s'exécute en parallèle de notre code. Je traitais donc le résultat de la requête avant de la recevoir ce qui posait de nombreux problèmes. Afin de pallier cela, j'ai ajouté un écouteur d'événement qui ne se lance que quand la requête est finie. Ensuite, en fonction du statut de la requête, on lance la suite du code ou pas. En effet, une requête HTTP émet toujours un statut qui permet de savoir si elle a abouti avec succès ou pas. Si elle a produit une erreur, cela ne sert à rien de continuer le code car cela créera plus d'erreurs qu'autres choses. On a donc bien vérifié l'existence de nos électeurs.

Il reste ensuite à vérifier que l'électeur n'est pas déjà présent dans la liste. Pour cela, il faut décider de quel type de données on met dans la chaîne contenant les électeurs que doit envoyer le formulaire. J'ai choisi les pseudos mais cela n'a pas vraiment d'importance. J'ai ensuite modifié l'API pour qu'à la place de renvoyer vrai, elle renvoie le pseudo de la personne. Néanmoins, si l'utilisateur n'existe pas, elle renvoie toujours faux. Ensuite, avant de lancer la fonction qui ajoute un électeur, on vérifie s'il est déjà présent dans la chaîne. Si c'est le cas, on l'ajoute pas et on prévient l'utilisateur sinon on l'ajoute.

3- Page du profil

Sur cette page, il est important que l'utilisateur puisse retrouver en quelque sorte l'historique de ses activités, c'est-à-dire gérer les sondages qu'il a créé et voir à quel sondage il a voté.

Pour les sondages auxquels il a participé, j'ai simplement récupéré dans la table "resultats" de la base de données les lignes où son pseudo était. Pour ceux qu'il a créé, j'ai fait de même dans la table "sondage".

Dans le but de gérer ses sondages, l'utilisateur va avoir besoin de voir les résultats. J'ai donc créé un bouton pour cela qui clôture le sondage et renvoie sur la page de résultat. J'ai aussi créé un bouton afin de supprimer le sondage.

4- Page de vote

C'est la page qui doit être la plus sécurisée car elle permet de saboter les votes. Pour cela, on récupère l'identifiant du sondage. On vérifie qu'il existe, qu'il est encore ouvert, que l'utilisateur n'a pas déjà voté. Ensuite, si le sondage est privé, on vérifie que l'utilisateur est sur la liste des électeurs. Si c'est le cas, il peut voter.

Sur cette page, il suffit d'afficher les informations contenues dans la base de données.On a juste ajouté l'attribut "required" à chaque groupe de bouton radio afin de s'assurer que l'utilisateur a sélectionné une réponse à chaque question.

Quand le formulaire est envoyé, on traite les données et on les enregistre dans la base de données.

5- Page de résultats

Dans un premier temps, on regarde si le sondage est fermé. Si ce n'est pas le cas, on le clôture car quand on regarde les résultats d'un sondage, ça veut dire qu'on accepte plus de votes.

Ici, j'ai choisi d'afficher dans un premier temps d'afficher des données importantes du sondage. J'ai donc affiché le nombre de votes obtenus, le nombre de votes attendu si le sondage est privé. On affiche également la date de création et de clôture.

Ensuite, on récupère toutes les réponses pour chaque question du sondage. On identifie les réponses avec la correspondance de leur identifiant dans la table "reponse". Il ne reste plus qu'à déterminer le pourcentage pour chaque réponse car c'est ces données qu'on utilise pour tracer le graphique avec Plotly. On a choisi de faire un diagramme circulaire car c'est le graphique le plus simple à comprendre et le plus visuel.

Retour sur le cahier des charges

L'application fonctionne bien. Néanmoins, elle a quelques défauts. Tout d'abord, elle n'est esthétiquement pas agréable du tout. Quand on supprime toutes les réponses et qu'on en ajoute des nouvelles, cela crée des lignes vides dans la table "reponse-possible".

De plus, tous le cahier des charges n'a pas été rempli.

En effet, premièrement, on peut créer un sondage avec moins de 2 réponses possibles par question voir aucune. Cela est réglable en ajoutant un attribut "onsubmit" au formulaire auquel on associe une fonction qui vérifie qu'il y a au moins 3 lignes(1 pour chaque réponse et celle pour en ajouter de nouvelles) dans chaque tableau de réponse.

Il manque également la fonction qui permet d'ajouter des questions lors de la création du sondage bien que le cas de plusieurs questions ait été traité dans la page de vote et de résultats.

Lors de la création d'un sondage privé, un mail devrait être envoyé à tous les électeurs. Or, comme le projet ne doit pas dépendre d'Internet pour aucune fonctionnalité, celle-ci a été impossible à mettre en place.

Sur la page du profil, dans chaque tableau, un colonne devait permettre de sélectionner des lignes mais comme les boutons ayant des effets sont directement dans la ligne du sondage correspondant, cela n'avait plus aucune utilité.

Sur cette même page, j'avais préparé tous les boutons mais je n'ai pas eu le temps de créer toutes les fonctions leur étant associées. Le bouton pour modifier la visibilité ne devait être disponible que pour les sondages privés car l'inverse aurait posé des problèmes de confidentialité. Le bouton pour modifier la date de clôture ne doit être disponible que pour les sondages ouverts. Le bouton afficher la participation devait afficher le nombre de votes obtenus pour un sondage.

Au niveau de la page des résultats, je n'ai pas eu le temps de rendre les graphiques copiables en cliquant sur un bouton. De plus, je n'ai pas affiché la durée d'un sondage car il ne s'agit pas d'une information très importante.

Licence du code

Comme j'ai utilisé Plotly, Tabulator et W3.CSS, la licence du code va dépendre de leur licence. Plotly a une licence MIT qui est une licence avec très peu de limitation. C'est donc une licence libre. Tabulator a la même licence. Il en est également de même pour W3.CSS.

De plus, comme il s'agit d'un site web, la licence du code serait une licence flottante. En effet, quelqu'un voulant accéder au site par internet se verra attribuer une licence par un serveur de licence pour un temps donné afin d'utiliser le site. Quand il a fini d'utiliser le site, la licence est récupérée par le serveur. C'est donc le modèle qui convient le mieux.

CONCLUSION

Durant ce projet, j'ai découvert les frameworks et comment les utiliser même si le résultat n'est pas visible sur le projet par manque de temps. En effet, j'ai très mal gérer mon temps donc j'ai privilégié les fonctionnalités qui me semblaient le plus importantes. J'aurais également appris que lors de la création d'un cahier des charges, il faut mieux l'alléger et trier les fonctionnalités par priorité. De plus, lors de la planification des tâches, il faut prévoir du temps supplémentaire pour gérer les problèmes qui surviendront.

D'un point de vue technique, j'ai appris comment mélanger ces cinq langages et comment faire passer mes informations entre eux. J'ai également découvert comment gérer et administrer une base de données du début à la fin, ce qui a été une grande découverte pour moi.

Néanmoins, ce module a été une déception pour moi car je n'ai pas été plus motivé que ça par le projet ni par les contenus des cours que j'ai passés à aider autour de moi et non à apprendre de nouvelles informations.