

# Rapport SAE 3.05

*Application web pour la gestion des activités de l'association*

## I. Poste et Mission :

Lors de cette SAE j'avais le poste de développeur Front-End où ma principale responsabilité était de réaliser les maquettes conçues au préalable sur Figma, mon rôle comprenait également la mise en place de la partie responsive de nos différentes pages pour garantir une adaptation optimale à tous les supports.

L'un des défis majeurs auxquels nous avons été confrontés était de rendre notre application web ergonomique pour faciliter l'utilisation de cette application par notre prestataire. Cela a nécessité une réflexion approfondie lors de la conception des maquettes en veillant à ce que l'expérience utilisateur soit intuitive et efficace. De plus, nous devions trouver un moyen de rendre l'application fonctionnelle dans les délais impartis.

Ma contribution s'est étendue à la liaison de la base de données produites par les développeurs Back-End aux différentes pages que j'ai développées. Au début de la SAE, j'ai également participé à la conception de la base de données sur Draw.io. Ensemble, nous avons imaginé une structure qui permettrait de stocker de manière efficace l'ensemble des contenus requis par notre client, tout en évitant de surcharger nos développeurs Back-End.

## II. Fonctionnalités implémentées :

Lors de cette SAE j'ai pu implémenter des fonctionnalités en JS :

- `openNav()` : Permet d'ouvrir une sidebar dans les pages de notre application
  - **Temps prévu** : 20 min
  - **Temps passé** : 40 min
  - **Problème rencontré** : Lors de la mise en place de la fonctionnalité d'ouverture de la sidebar, j'ai rencontré plusieurs difficultés initiales en tentant de le réaliser en CSS. Cependant, face à ces obstacles, j'ai décidé de m'aventurer dans un langage qui m'était jusqu'alors inconnu. J'ai décidé d'utiliser le langage Javascript pour résoudre ce problème. Bien que cela représentait un défi, j'ai estimé que c'était la meilleure option pour résoudre ce problème spécifique. Le principal obstacle résidait dans le fait que je n'avais aucune connaissance préalable du langage JavaScript. Pour surmonter cette lacune, j'ai consacré une journée entière à l'apprentissage des bases de ce langage. Pour ce faire, j'ai utilisé la formation disponible sur OpenClassroom, consulté la documentation détaillée de MDN Web Docs et effectué des exercices pratiques sur Exercism. Cette approche m'a permis de rapidement m'approprier les différentes syntaxes et concepts fondamentaux de JavaScript. Une fois que j'ai acquis une compréhension suffisante du langage, j'ai pu développer la fonction nécessaire pour ouvrir la sidebar. J'ai également veillé à factoriser le code pour le rendre plus lisible. Ce processus m'a non seulement permis de résoudre un défi technique spécifique, mais il a également renforcé mes compétences en programmation JavaScript, élargissant ainsi ma palette d'outils dans le domaine du développement web.
  - Sources utilisées :
    - <https://openclassrooms.com/fr/courses/7696886-apprenez-a-programmer-avec-javascript>

- <https://exercism.org/tracks/javascript/concepts>
- <https://developer.mozilla.org/fr/docs/Web/JavaScript>

- `closeNav()` : Permet de fermer une sidebar dans les pages de notre application

- **Temps prévu** : 30 min
- **Temps passé** : 35 min
- **Problème rencontrés** : La mise en place de la fonctionnalité de fermeture de la sidebar s'est déroulée sans difficultés majeures. J'ai simplement dû apprendre à créer une fonctionnalité permettant d'ajouter un délai lors de la fermeture des pages. Pour faire cela, j'ai exploré les fonctionnalités de délai disponibles en JavaScript. Puis j'ai utilisé la fonction `setTimeout()` qui permet d'exécuter une fonction après un certain délai spécifié.
- Sources utilisées :
  - <https://developer.mozilla.org/fr/docs/Web/JavaScript>

- *Un écouteur d'événement pour les clics* : Permet de fermer la sidebar lorsque l'on clique sur un endroit en-dehors de la sidebar

- **Temps prévu** : 10 min
- **Temps passé** : 1 h
- **Problème rencontrés** : La mise en place de l'écouteur d'événements s'est déroulée sans encombre, mais j'ai consacré une quantité significative de temps à résoudre un problème spécifique lié à la condition. La difficulté résidait dans le fait que je n'avais pas initialement pensé à vérifier le type des variables, ce qui a entraîné des erreurs de comparaison. Pour résoudre ce problème, j'ai utilisé l'opérateur `typeof` pour vérifier le type de mes variables. Cela m'a permis d'identifier le type de données avec lequel je travaillais.
- **Source utilisées** :
  - <https://developer.mozilla.org/fr/docs/Web/JavaScript>
  - <https://openclassrooms.com/fr/courses/7696886-apprenez-a-programmer-avec-javascript>

- `afficher()` : Permet d'afficher une partie des statistiques qui est cachée lorsque que l'on est sur la page Statistique

- **Temps prévu** : 10 min
- **Temps passée** : 5 min
- **Problème rencontrés** : je n'ai rencontrés aucun problème lors de l'élaboration de cette fonction
- **Source utilisées** :
  - <https://developer.mozilla.org/fr/docs/Web/JavaScript>

J'ai systématiquement intégré des instructions `console.log()` dans toutes mes fonctions pour faciliter le débogage de mon code. Cette approche m'a permis d'afficher les valeurs des variables ainsi que des informations cruciales sur l'exécution des fonctions dans la console du navigateur. Grâce à cette pratique, j'ai pu suivre efficacement le déroulement du code, repérer les erreurs éventuelles et garantir ainsi la fiabilité de mon programme.