|  |
| --- |
| Collège de Bois-de-Boulogne |
| Journal de Anis TP2 |
| 420-2CW Applications Web |

|  |
| --- |
| Hamiteche, Anis Hani  21/05/2024 |

Table des matières

[Défis rencontrés 3](#_Toc167398804)

[Solutions trouvées 4](#_Toc167398805)

[Apprentissage 4](#_Toc167398806)

[Les captures d’écran du *git reflog* 5](#_Toc167398807)

# Défis rencontrés

Lors de l’exécution du travail pratique, l’expérience a été enrichissante et intéressante. J’ai pu mettre en évidence la matière acquise dans un cadre pragmatique, afin d’avoir une idée de ce qu’est créer, gérer et intégrer une base de données dans un site web pour le rendre dynamique et plus accessible. Toutefois, il y a eu quelques obstacles qui m’ont non seulement rendu la tâche plus complexe, mais aussi plus lente, car cela a pris plus de temps à les résoudre. En mentionnant seulement les plus marquants, je m’explique en chronologie :

Le premier est que j’ai commencé plus tard que ce qui a été prévu. En fait, j’ai pensé débuter trois semaines plus tôt que la date de remise, puisque le TP a eu l’air d’être long. En final, le début s’est fait deux semaines à l’avance. Donc, j’ai perdu une bonne semaine – qui était beaucoup plus libre qu’après, de temps.

La tâche a été divisé pour rendre le travail plus rapide et efficace : moi m’occupant de la création de la base de données et Kimi de la formation et du design du site web. Thème et tables décidés, j’ai fait ma base de données en mettant les éléments pour former les colonnes sur *Data Modeler*. Au moment de mettre les relations, ça a été un peu plus haut en difficulté, car j’ai été indécis sur la logique. Par exemple, pour la table Clients et Réservations, ça a été comme ça : « Est-ce qu’un client peut seulement faire une réservation ou bien plusieurs? Si c’est le premier cas, ça sera une relation 1-1 et si c’est le deuxième, alors 1-N sera la réponse. » En d’autres mots, la relation entre les tables m’a pris un bon bout de temps à appliquer, ce qui a en partie affecté une partie de la progression.

Pour quelques raisons, durant une fin de semaine, mon coéquipier n’a pas eu le temps de commencer le site web. Alors, j’ai pris le relais. En général, ça a été intéressant et tranquille, jusqu’à la formation des… boutons. Oui, puisque je voulais en faire des dynamiques et des beaux en les plaçant correctement, ça m’a pris une à deux bonnes heures. En effet, la cause n’a pas été en codant les boutons, mais bien en faisant des essais et des erreurs. Parfois, certains styles ne me convenaient pas. Donc, je refaisais chaque fois le code CSS jusqu’à avoir la bonne. Et ça, ça a été un des obstacles majeurs de la progression et du temps.

Le dernier est le plus marquant : l’affichage des données sur une page du site web. Que vous me croyiez ou non, ça m’a pris une journée entière à afficher les données de ma BDD locale en codant sur JavaScript et HTML. Soit j’ai eu des erreurs, soit des codes infonctionnels. Pourquoi m’a-t-il pris tout un jour à seulement afficher des données sur un fichu site web stocké sur mon PC? Eh bien, la réponse est simplement qu’initialement, j’ai pensé qu’il fallait faire une page où l’utilisateur remplit des champs (en faisant des <input></input>) pour que les données saisies s’enregistrent sur ma base de données. Cette partie a réservé près la demi-journée de mon temps. Après que j’ai su que ce n’était pas ce qu’il fallait réaliser, j’ai tenté de faire le code JavaScript et HTML pour relier mon lien API de ma base de données au site en précisant les éléments à afficher : ça n’a pas marché. C’était sûrement une erreur. Quelques modifications faites : ça n’a pas marché. J’ai refait un autre code : ça n’a pas marché. J’ai tout vérifié et refait à la lettre chaque ligne sur les deux langages : ça n’a pas marché. J’ai fait appel à une aide externe : ça n’a pas marché. Bon sang! Où a été le problème? Après une bonne autre demi-journée, j’ai enfin réussi à le trouver et à afficher mes quatre tables sur la page en faisant le bon code!

# Solutions trouvées

J’ai eu des problèmes et des solutions pour former le site web dynamique. Félicitations! Mais, comment?

Celle du premier obstacle a été réglé facilement et rapidement. Avec Kimi, on a littéralement commencé immédiatement lorsqu’on s’est mis ensemble pour le projet pour gagner plus de temps.

La deuxième a été de plus réfléchir et de suivre l’idée la plus logique pour mettre les relations en place entre les tables.

La troisième a été de corriger des erreurs dans le code jusqu’à trouver le style le plus satisfaisant.

La dernière a été de faire appel à une aide externe pour trouver le bon code et à moi de le former correctement pour l’ajuster à la situation.

# Apprentissage

Qu’est-ce que j’ai appris lors du travail pratique? Telle est l’ultime question à poser à la fin ce projet collectif. Tout au long du parcours, j’ai appris beaucoup et voici la brève liste :

D’abord, j’ai eu l’occasion de me refamiliariser avec HTML et CSS, à me souvenir de certaines options et fonctionnalités et à coder le squelette d’un site web intéressant. J’ai pu rendre ma performance de programmation plus professionnelle, propice et efficace. Je ne vais pas aussi cacher que j’ai appris quelques nouvelles balises que je n’ai jamais utilisées auparavant. Par exemple, sur HTML, <input> n’a jamais été l’une de mes pratiques, celle qui permet de mettre une case blanche où l’utilisateur peut saisir du texte. Aujourd’hui, je pense que c’est un élément crucial à la formation d’un site web dynamique.

Ensuite, j’ai eu l’opportunité d’apprendre le langage JavaScript et d’enrichir ma connaissance là-dessus. Avant, la seule balise que je connaissais était <script> et la méthode alert(). Je les utilisais pour seulement avoir une idée très générale de ce qu’est ce langage. Maintenant, je sais non seulement faire des boîtes d’alerte, mais aussi relier une base de données locale à un site web fait avec HTML! C’est très *cool*.

De plus, j’ai appris à coder en SQL et à utiliser les quatre énoncés principaux de ce langage : manipulation, définition, contrôle et transaction. Ceux-ci m’ont permis de créer une base de données efficace et basiquement sécurisé.

# Les captures d’écran du *git reflog*

Cette section contient les captures d’écran de toute mon activité faite jusqu’à présent sur ma branche « anis » dans le répertoire du TP. La dernière consiste à une autre commande *reflog* sur le nouveau répertoire (actuel), après une migration de notre projet de l’ancien à celui d’aujourd’hui pour s’installer sur l’équipe de votre *classroom*. Les voici :

Une image contenant texte, capture d’écran

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, capture d’écran

Description générée automatiquement