

Rapport de stage BTS1 S.I.O S.L.A.M: Mameri Heddy

Date : 3 au 28 Juin 2024

Maître de stage : Charles Haller

Introduction :

Dans le cadre de la formation en BTS nous devons réaliser 10 semaines de stage en entreprise, nous en faisons donc 4 la première année et 8 à la seconde année. Le but de ce stage est de gagner en expérience et de constater comment se déroule le travail en entreprise. À ce titre l'entreprise doit nous donner un projet à faire qui nous demandera une amélioration de nos compétences et de nos connaissances grâce à l'expérience des développeurs autour de nous.

Pour ce stage première année, je suis allé dans l'entreprise EC2E. Leur cœur de métier est la manutention. Ils proposent à leur client des applications et matériels pour superviser et gérer les engins industrielles via un système de badge. Ils proposent également le développement de centrale de paiement pour laverie automatique, que se soit pour le fonctionnement ou la sécurité. En tant que stagiaire de BTS1 SIO SLAM, on rejoint le service Recherche & Développement de l'entreprise. Étant donné que mon niveau n'est que de première année, on ne met pas sur un projet client mais sur une petite application interne que l'équipe R&D pourra utiliser. Le projet que l'on m'a donné est «heure_développement».

Contexte : Chaque année, les membres du service R&D doivent remplir un tableau excel pour y mettre leur nombre d'heures de travail pour chaque jour de l'année week-end et jour férié exclu. S'il ne travaille pas ils doivent préciser leur occupation (congé non payé, congé payé, arrêt maladie, ...), s'ils travaillent ils doivent préciser sur quel projet et les détails du travail de la journée. Le but est d'avoir le total d'heure effectué pour calculer un taux pour les impôts. Ma mission est de créer une application pour remplacer ce tableau excel.

méthode a utilisé qui m'ont amené à faire des veilles et de l'auto formation dessus.

Mon application doit comporter 4 onglets : - Accueil avec un tableau de vue d'ensemble où chaque ligne correspond à un jour et chaque colonne correspond à un développeur, on doit pouvoir y ajouter des journées.

- Recherche, permettant de rechercher un employé et retourner un tableau plus précis cibler sur ce même employé avec la possibilité d'ajouter des informations sur une journée et de les modifier. Mais aussi de faire des recherches par projets occupations et années.

- Statistiques, affichant divers données tel que le nombre d'heures total par année, le nombre d'heure total d'un employé sur tel année, le nombre d'heure total par employé sur les 4 trimestres d'une année, et le nombre d'heure total que les employés ont effectué sur toutes les années.

J'ai également plusieurs contraintes techniques mis en place au fur et à mesure de

l'avancée du développement :

- Mise en place d'une architecture MVC.
- Utilisation de bootstrap version 5.3 pour gagner du temps en front
- Utilisation de ChartJs
- Utilisation de requêtes AJAX pour dynamiser l'application.

La sécurité via une page de connexion demandant un identifiant mot de passe n'est pas nécessaire car je travaille sur une application pour une équipe en interne. On ne m'a donc pas demandé d'en faire.

On ne me demande pas non plus de créer un répertoire github, avec des issues, milestone, ...

Dérouler du projet :

1ere semaine :

Lors de la première semaine, on ne commence pas le développement de l'application mais on réalise une maquette Figma pour définir la direction à suivre. M. Haller demande à ce que j'étudie l'architecture MVC qui m'était inconnue avant. Le MVC consiste à séparer les différentes fonctionnalités et parties de code d'une application en 3 répertoire. Le principe est d'avoir une seule page «index.php» pour tout afficher qui va récupérer les données via des require_once. Grâce à un fichier htaccess on va modifier l'URL du site pour quelque chose de compréhensif par l'utilisateur. C'est là qu'interviennent les 3 répertoires. Views (vue) contient tout ce que le site va afficher tel que les formulaires, tableaux, ...), models contient des classes correspondant aux tables de la base de données et les méthodes de ces classes à des requêtes SQL qui vont appliquer le CRUD (Create Read Update Delete) mais aussi récupérer les données qui nous intéressent. Enfin le controller fait le lien entre les views et les models, il récupère les données de formulaire pour les envoyer au model après vérification. Le controller contient des fonctions qui vont servir à retourner les données correspondantes pour chaque views, donc chaque views a sa fonction dans le controller. Il y a un controller par model et c'est également ce dernier qui fait les require_once des views pour les afficher dans l'index. Grâce à cette méthode on obtient l'url «nom_du_projet/nom_controller/nom_fonction_controller/» . Afin de maîtriser l'architecture MVC, il est mieux de réaliser un travail pratique dessus pendant la première semaine. Ce TP permet juste d'administrer une liste de personnages avec leur nom , prénom, âge,... Il m'est un peu difficile de comprendre la page index.php se trouvant dans le cours avec la définition du router et les conditions pour accéder au contrôleur. De plus les modèles sont des classes donc il faut connaître la syntaxe du php objet, c'était assez simple car similaire au C# objet. J'ai facilement réussi à créer des classes, les vues aussi. Cependant j'ai eu des difficultés à permettre à une vue d'appeler un contrôleur. En fait il faut mettre l'url type MVC dans le action des balises form. La seconde difficulté que j'ai rencontré liée au MVC était les redirections, car quand on cliquait sur un bouton le nom du contrôleur et de sa fonction se dupliquait dans l'url. Lorsque j'essayais de mettre un

header Location ça ne marchait pas car les redirections doivent être mis dans des if pouvant précéder d'autres lignes de code. La solution est de faire echo sur des redirections js windows location. Une fois tous ces problèmes résolus le développement de l'application a commencé.

2e semaine :

Lors de cette semaine j'ai développé les différents onglets en commençant par l'administration, étant donné que j'avais déjà fait celle de TP_MVC. Ensuite j'ai fait le menu et le tableau de la vue d'ensemble, cependant il y avait des problèmes de front avec des divs qui rentraient en collision. On m'a dit d'utiliser bootstrap pour ça. M. Haller m'a expliqué les classes col et row de bootstrap, comment aligner et régler la taille que prend une col dans une row. Ensuite il m'a montré une documentation bootstrap en ligne afin d'avoir des modèles. J'ai trouvé les boutons qui ouvrent des fenêtres modales et j'ai eu l'idée de les utiliser pour le menu et les formulaires de l'application. Grâce à cela il n'y a plus de problèmes de divs qui s'entrechoquent.

Pour le tableau de la vue d'ensemble j'ai eu besoin d'aide car il fallait mettre tous les jours de l'année avec des conditions. J'ai demandé à chatgpt de me fournir une fonction pour générer ce tableau, un alternant m'a aidé à faire les conditions. Le problème est que je ne savais pas quoi mettre. Il a eu l'idée de comparer les jours de les jours enregistrés dans la base de données avec ceux du tableau php allDays.

J'ai ensuite fait l'onglet statistique, cependant je ne savais pas comment faire de canvas. J'ai demandé à chatgpt, et il m'a expliqué qu'on en génère avec ChartJs. J'ai essayé et ça a marché.

Enfin la plus grande difficulté de cette semaine est l'incrémentation automatique des heures dans le formulaire d'ajout d'une journée. Tout le problème réside dans le fait que le formulaire concerné se trouve hors de la boucle foreach qui affiche les données. Bien qu'à la base mes connaissances sur javascript se résument à savoir faire des carousel et changer du style, j'ai fait des recherches sur javascript et j'ai trouvé l'attribut data. Grâce à un data-nom_attribut dans le bouton qui appelle le formulaire, on peut avoir les infos de la ligne concernée en récupérant les données et en les attribuant comme valeur d'identifiant javascript.

J'ai enfin fait l'onglet recherche et le tableau associé. La problématique était qu'il fallait griser les lignes du week-end, mais avec une recherche de la valeur numérique des jours de 0 à 6 ça c'est arrangé. J'ai corrigé un problème d'affiche lié à la boucle du tableau des jours. Lorsqu'un utilisateur n'avait pas de détails et de nombre d'heures parce-qu'il était en congé, ..., la div de l'occupation s'affichait sur toute la ligne. J'ai arrangé le problème avec une condition if else et un display none si l'occupation n'est pas travail ou télétravail.

3e semaine :

Lors de la 3e semaine je devais faire des modifications afin que l'application puisse s'adapter à l'année actuelle et catégoriser les données de statistiques par année. J'ai fait des recherches pour isoler les années des dates de ma table sql. J'ai trouvé la

fonction sql YEAR pour cela et même QUARTER pour les trimestres d'une année. J'ai rajouté des boucles pour qu'un canvas se génère par année. J'ai mis une navbar bootstrap sur les formulaires de l'onglet recherche pour faire plus beau.

Le problème avec l'adaptation par année était de s'adapter à l'année actuel de l'ordinateur, et surtout de conserver cette valeur en filtre de la barre de recherche de l'onglet recherche. J'ai créer une variable year et lui ai mis par défaut la valeur de l'année actuel. Conserver les filtres de recherches n'était pas facile non plus. J'ai demandé à chatgpt comment faire et il m'a montré les variables de sessions. J'ai testé et ça a fonctionné. Ces mêmes variables ont été utile pour les incrémentations de valeurs automatiques dans le formulaire tel que les id utilisateur,... Les tests pour vérifier si cette fonctionnalité marche ne sont pas très commun car il demande de changer la date interne de l'ordinateur dans les paramètres.

J'ai arrangé le front de cette page pour que tous les tableaux prennent pour place l'ensemble de la page, très facile avec bootstrap. J'ai ajouté des min-width aux cellules du tableau de vue d'ensemble car le nombre des colonnes est le nombre d'employé du service R&D actuellement à huit. Il y a eu des problèmes avec les min width parce-que certaines lignes de code qu'on avait importé pour une scroll bar sur le tableau rentrait en conflit avec le min-width, heureusement je l'ai trouvé et corrigé. Grâce à cela toutes les cellules sont responsive et ne peuvent voir leurs divs faire des bugs de front à cause de la taille de l'écran. J'ai fait la fonction searchDetail pour faire des recherches dans le tableau et ne retourner un tableau qu'avec les résultats via des requêtes select. La difficulté est de prendre en compte que l'utilisateur peut rechercher un projet sans occupation, une occupation sans projet ou une occupation et un projet. Pour gérer ça j'ai mis des conditions if pour tester si les superglobales sont vides et si oui lesquelles. Enfin la recherche ne peut-être valider que si deux variables verif ont pour valeur vra. Pour quelles soient vrai il faut que les superglobales POST user_id, date, et que occupancy , occupancy_id existent et contiennent des données en plus qu'il ne faut pas que la superglobale projet soit compléter mais que l'occupation soit égal à travail ou télétravail. Pour l'affichage du résultat de recherche j'ai aussi penser à créer une classe css pour mettre un background-color yellow.

4e semaine :

En voyant mon site sur la dernière semaine, M. Haller a demandé des modifications de front pour rendre le tout plus homogène est agréable visuellement, des shadow-box pour rendre des éléments plus lisible, et des alignements d'éléments comme les cards de administrations pour homogénéiser le rendu visuel. Mettre des background-color pour bien différencier des tableaux ou canvas du fond du site. Aligner des cards dans administration pour rajouter de la cohérence.

Pour la dernière semaine, on m'a demandé de faire des requêtes ajax sur les formulaires de tableaux d'année afin que lorsqu'on ajoute ou modifie une information la page ne se recharge pas. L'intérêt est d'éviter qu'à chaque ajout ou modification l'utilisateur ne soit rediriger vers le haut du tableau. Ces requêtes ont

posé beaucoup de problèmes. Je fait des petits formulaires pour comprendre l'ajax notamment la récupération des informations. Toute la difficulté à résider dans le fait qu'il fallait envoyer les valeurs récupérer à la page pour afficher les informations par exemple le nom de l'occupation ou du projet, la couleur de l'occupation, les détails,... Résoudre ce problème a duré une après-midi entière avec l'alternant. J'ai appris à debugger en javascript avec console.log. En mettant des id au td du tableau et faisant des echo js des divs, la requête ajax a finalement fonctionné pour l'ajout de journée dans l'onglet recherche. Une petite subtilité pour les requêtes ajax est de mettre des valeurs par défaut sans que l'utilisateur ne touche au formulaire, c'est important pour projet et détails, car il y a l'option aucun.

Cependant, il ne marche pas pour modifier car il ne récupère pas toutes les valeurs de utilisateur.

Le formulaire de recherche de projet et d'occupation ne marche plus parce-que j'ai essayé de le fusionner au formulaire de recherche d'un utilisateur en voulant rendre ça plus simple car normalement il y a 2 formulaires et 2 boutons rechercher dans la nav bar.

Présentation :

Pour ce qui est de la présentation de mon projet . Seuls manquaient la recherche de projet et d'occupation lié à la mise en commun du formulaire de la side bar et la modification de journée lié aux difficultés d'AJAX. Sinon les retours de l'équipe ont été assez positif. Ils disent que c'est très bien pour un élève de première année. Que d'une certaine manière j'avais surpassé les attentes. À la fin j'ai remercié tout le monde de m'avoir accueilli dans l'entreprise.

Conclusion :

En conclusion ce stage aura été extrêmement instructif pour moi, je suis assez fier de moi car j'ai créer une application qu'une équipe de développeurs va reprendre pour l'utiliser.

Pour ce que Charles pense de moi il m'a trouvé autonome, très professionnel, poli mais un peu trop réservé. Il me recommande d'aller vers les autres.

J'ai adoré mon stage à EC2E, ce dernier a été très instructif et m'a permis de prendre un peu d'avance sur l'année prochaine. J'ai fait du PHP objet ,j'ai étudié bootstrap pour gagner du temps en front. J'ai aussi vu ajax pour dynamiser un site. On m'a fait la remarque que le projet a réalisé n'était pas très facile parce-que j'ai tout coder depuis zero en ce qui concerne le back end, alors que les framework sont censés automatisés ça ce qui a rajouté de la difficulté mais d'un autre côté je pourrai mieux comprendre l'intérêt de les utiliser.

Points forts :

- Travail sérieux
- Autonomie : je cherchais toujours sur Internet les solutions aux problèmes avant de demander de l'aide car tout le monde est débordé.
- j'ai montré l'application avec la quasi totalité des fonctionnalités attendus.
- l'équipe a trouvé que mon projet était très bien pour un niveau de première année.
- J'ai pris de l'avance sur les années à venir. Que se soit Symfony qui automatise le MVC ou VueJS qui automatise les requêtes AJAX. Je sais que se ne sont pas pas tous les stages de première année qui nous permettent de nous améliorer en tant que développeur, j'ai conscience d'avoir eu de la chance grâce à l'aide de M. AMMAR et de Charles HALLER. Même si je suis resté assez réservé je pense m'être plutôt bien intégrer dans l'entreprise, tout le monde a été très gentil. Pour ce qui est du projet, je ne l'ai pas trouvé très difficile ne lui-même mais le problème principal résidait dans le fait que Charles m'ajoutait des contraintes techniques au fur et à mesure que je le développais ce qui a posé problème à la fin car je n'ai pas eu le temps de faire marcher l'ajax sur la modification du journée.

Points à améliorer :**- compétences :**

- être plus ouvert aux autres, dans des conversations du quotidien avec ses collègues.

- projet :

- corriger le formulaire de recherche soit en mettant une condition de la fonction d'affichage par utilisateur pour garder un formulaire doit le re diviser en deux.
- corriger la requête ajax de modification, faire des console.log, trouver pourquoi certaines informations ne s'envoient pas.
- mettre les theads des tableaux en position fixed pour que l'utilisateur ne se perdent pas dans la vue d'ensemble.
- Corriger le front du tableau des heures d'employés par année car avec trop d'année la structure actuel va être illisible. Notamment le fond et les couleurs.
- Possiblement mettre une barre de fin avec conditions d'utilisations, rgpd,