Dev Mountains

Rapport de stage

Favre Sylvain

05/01/26

13/02/23

00 Remerciements

Je remercie

Rougeron Anthony pour son tutorat

Kevin Pietrovich et Jade Rapallini pour leur accueil à l’incubateur

Sébastien Marchand pour la mise en relation

Ainsi que l’ensemble des porteurs de projets de l’incubateur

01 Introduction

Ce stage à été effectué à l’entreprise DevMountains , sous la tutelle de Anthony Rougeron.

Elle est située au 16 rue Carnot, 05000 Gap, dans les locaux de l’incubateur GAAAP.

anthony@devmountains.fr
+33 6 32 26 79 85.

Référent : Sébastien Marchand.

02 L’entreprise

Micro-Entreprise crée en 2014 par Rougeron Anthony sous le statut d’entreprise individuelle (EI), spécialisée dans le développement informatique de sites internet s’appuyant principalement sur les stacks Symfony, Vue.js et Nuxt .
Elle est dirigée par son fondateur, unique dirigeant et salarié, présent dans les locaux de 8h30 à 17h (et au-delà selon les besoins de l’activité).

03 L’activité

Devmountains à pour principale activité le développement d’applications sous licence de type SaaS (Software as a Service).

Elle intervient également dans l’optimisation des processus internes ainsi que dans l’accompagnement numérique des entreprises.

Son objectif est le partenariat avec des entreprises, et le dévelopement d’applications créatives.

04 Les Finalités

La finalité économique de l’entreprise est de faire de l’argent en proposant des services ajoutant de la valeur aux processus, performances, et à l’efficacité de ses clients.

La finalité sociétale de l’entreprise est de fidéliser les salariés en favorisant le bien-être au travail, les avantages, et l’évolution professionnelle. Tout cela afin de maximiser leur investissement sur les projets.

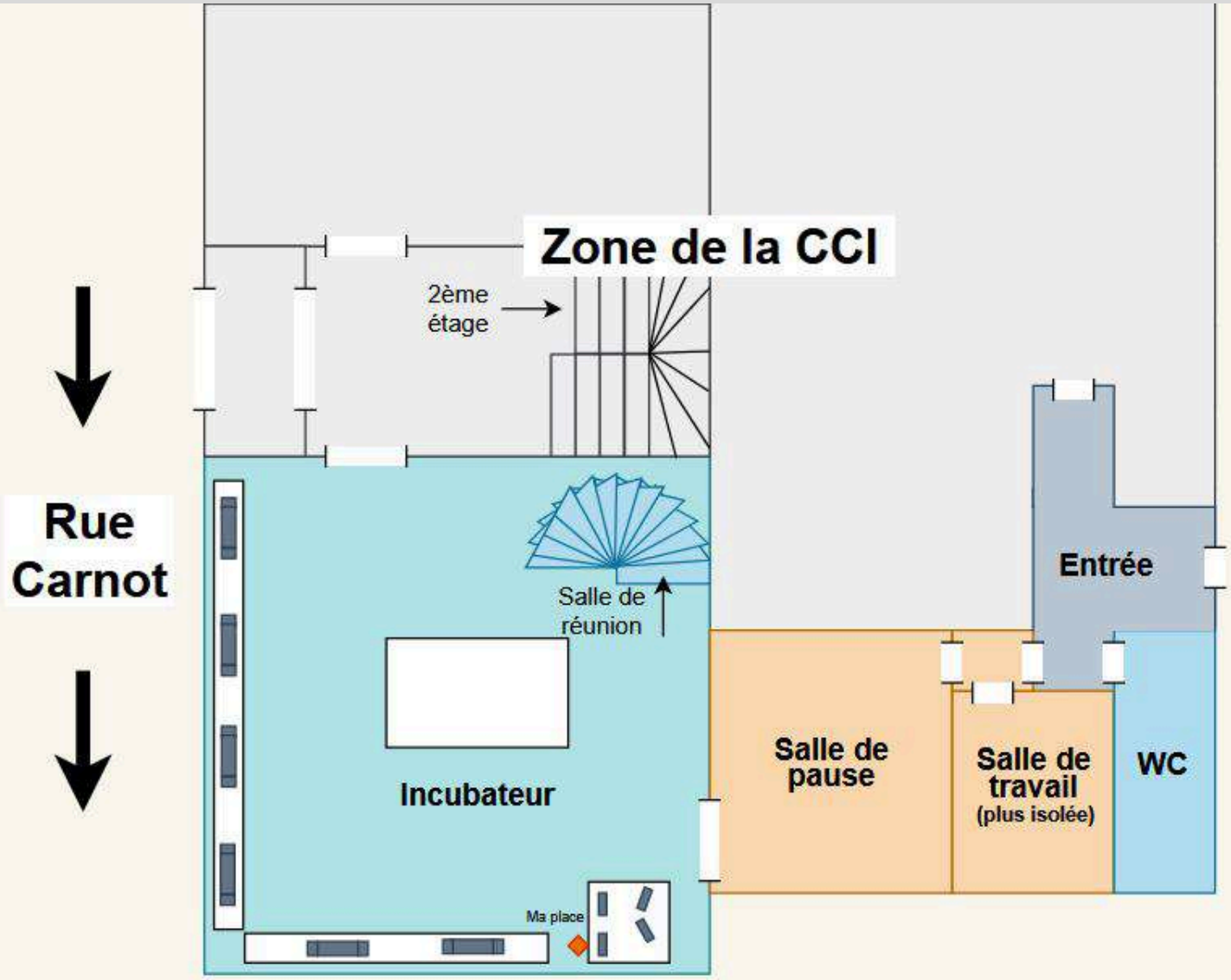
07 Outils

Environnement :
Stack WAMP (Windows, Apache, MySQL, PHP) pour la bdd MySQL, et framework Symfony.

Sites Internets :
GitLab pour l'hébergement du code, Notion pour organiser le workflow, et Trello pour planifier les tâches.

Outils supplémentaires :
VSCode pour le developpement, Google Drive pour les notes de projet et Obsidian pour le suivi des connaissances acquises.

08 Plan des locaux



09 Présentation du tuteur

Anthony Rougeron · 2e

Développeur Backend Symfony

Développeur Backend PHP

Je vous accompagne vers des API robustes et scalables

From Scratch Migrations Optimisation

Services

Développement web • Développement SaaS • Développement de logiciels personnalisés • Développement d'applications • Développement de bases de données

J'accompagne mes clients dans la création d'applications robustes et évolutives, que ce soit pour démarrer un projets from scratch ou en rejoignant leur équipe :

Passionné par l'apprentissage des métiers de mes clients, je m'implique pour comprendre en profondeur leur domaine d'activité.

Apprendre leurs défis et spécificités, pour proposer des solutions sur mesure qui collent vraiment à leurs besoins.

Accompagner mes clients, pour moi, ça va bien plus loin que coder : c'est s'assurer que chaque solution fasse sens pour leur activité.

Intervenant sur la stack backend, je peux :

- Développer une application ou une API orientées métier
- Effectuer des migrations de versions PHP & Symfony
- Mettre en place une Clean Architecture en m'appuyant sur les concepts de DDD, BDD et TDD
- Refactoring
- Rédiger la documentation technique

J’ai signé une clause de confidentialité. Cette deuxieme partie ne parlera donc pas du fond des tâches effectuées, mais de ce que j’ai vu et appris.

Travailler sur un même projet quand on est en équipe demande de la préparation.
Dans notre cas, le trello contient toute les features a coder , et qui doit les coder.
On clone le projet, puis on crée et travaille sur une branche feature/nom_feature mise à jour tout les matins avec la branche principale.
Une fois la feature complétée, on s’assure une nouvelle fois que la branche est à jour, puis on la met sur le repo distant de gitlab avant de créer une merge request qui mettra la feature sur la branche principale, détruisant la branche feature/nom_feature.

12 Refactorisation

En principe, on rend une partie du code réutilisable, en transformant tout ce qui est variable dans son éxecution en temps que paramètre. Mais dans une démarche professionnelle, j'ai appris à aller plus loin : séparer les métiers (chaque ligne de code avec une fonction précise mise dans son propre fichier, dans son propre dossier), adapter le code a la situation (ne pas transformer en classes ce qui peut être une structure de donnée, éviter l'over engineering,...) et faire du clean code (remplacer les commentaires par des noms de fonctions, entre autre).

On passe d'un sac a lettre fonctionnel, à un emboitement complexe mais propre/réutilisable/modulaire de fichiers.

13 Analyse Métier

A la fois régi par des lois strictes et des singularités propres au client, expliqué comme si on était dans la sa tete, et dont la topographie change avec les informations recues au compte goutte, j'ai appris qu'il est indispensable de se faire une idée du paysage dans lequel va évoluer notre outil avant de se jetter dans sa conception.

En temps qu'acteur de services informatiques prodigués aux organisations, j'ai appris qu'il ne faut pas juste faire ce que le client ordonne, car il ne sait pas ce qui est possible. Il faut apporter une expertise numérique adaptée aux besoins.

Cette analyse est passée par des tableurs excels, des notes de réunions prises à la hâte, des documents métiers difficiles a comprendre... qui aurait été plus facile si tout était préparé, complet, et bien formulé.

14 Analyse Logicielle

Le paradigme du fondateur de devmountains, c'est la dépendance inversée. Ne pas dépendre d'un service en particulier pour faire une tache, mais de n'importe quel service qui implémente les fonctions demandée par la tache.
J'ai pu voir son intêret et implémentation à travers le design pattern adapter (un genre de matriochka d'interfaces) souvent utilisé dans le logiciel. J'ai été témoin de formType/ repeaters (SymfonyForms vanilla > EasyAdmin) formant des cruds bien configurés.

L'étude en amont des composants, services, et designs patterns utilisés est complexe au début , mais facilite la localisation de l'origine des bugs/l'ajout de fonctionalités.
Je l'ai faite en cheminant le long des fonctions, méthodes, classes et interfaces correspondants à chaque étape du parcours utilisateur (en allant parfois trop loin).

Esprit Symfony

Symfony impose un cycle de développement précis : ne pas le respecter conduit à un code désorganisé, redondant et instable ; Tel Dorothée, il faut suivre le chemin de briques jaune.
En l'utilisant, j'ai pu comprendre que les symfonyForm sont très puissant si on les configure bien. Les méthodes de controlleur doivent être courtes, toute la logique métier se trouvant dans des services . On peut créer des entités dans des dossiers en mettant un bon nom . On utilise des attributs pour configurer la plupart des méthodes. Symfony possède un event listenner un peu comme js, mais pour des events spécifiques ou locaux.

16 Esprit Bootstrap

C’est bête a dire, mais je n’ai jamais autant utilisé et compris bootstrap que pendant ce stage. le pouvoir de configurer des éléments directement via les classes, afin de combiner les aspects ensembles, et très utile. Mais les composants natifs le sont aussi : des cards aux icônes, des couleurs primaires et breakpoints responsives aux composants plus complets comme le spyscroll, le front n’aura jamais été aussi simple.

Ca m’efforce a utiliser des éléments UI préconstruits plutot que de tout faire “from scratch” , ce que j’ai encore du mal a faire.

Consulting

J’ai eu le plaisir d’utiliser mon background en géologie pour le compte d’une porteuse de projet.
On m’a expliqué le projet, les technologies utilisées, et j’ai raconté mes experiences pertinentes et les pensées que j’ai eu sur le coup. Ca m’a permis de faire l’experience du client dans un projet : tout n’est pas clair dès le début, mais à force de questions et d’ouvertures, on arrive à certaines conclusions. Pour avoir été des deux côtés d’une réunion, je peut maintenant dire que la principale barrière sont les aprioris. Le coté métier présume que certaines choses sont logiques, le coté prestataire présume du fonctionnement de certains éléments, et on arrive à des problèmes mal identifiés et des produits mal adaptés.
Il faut être vigilant à ne pas prendre certaines connaissances pour acquises.

18 Problèmes/Solutions

J’ai eu du mal a faire marcher les repeaters imbriqués pour les formType. Je me suis trop reposé sur gpt pour trouver le probleme, mais ce que je voulais coder était trop complexe pour lui, et j’aurais résolu le problème plus vite si je m’étais vraiment penché sur le fonctionnement des thêmes customs.
J’était en train de recoder doctrine pour une entité parce que je n’ai pas “suivi le chemin de brique jaune” (symfony peut lier autant de formulaires à une entité, et, bien lié, il suffit de persister l’entité pour que ca fasse tout les changement necessaires dans la bdd).
Au final, mes principaux problemes, c’est “utilise encore trop l’IA” et “se jette dans le code sans chercher de moyens pré existants”.
Cepandant, j’ai plus utilisé les documentations que l’IA (comme promis dans mon dernier rappport), et j’ai noté mes trouvailles dans mon obsidian, une pratique que je pense continuer dans le futur.

19 Conclusion

Ce stage aura été pour moi une destruction localisée de l'ego. J'ai du apprendre que je ne savais rien, que je ne valais rien, que je n'avais même pas atteint la ligne de départ. J'ai du voir que mes efforts n'étais pas suffisants, me séparer de mes habitudes, mettre la distance entre moi et mon code. Tel Einstein devant l'univers, la seule chose que je sais, c'est que je ne sais rien. On m'a jeté à terre, et on m'a montré ce qu'il fallait faire pour se relever. Je n'ai pas appris a coder en symfony, j'ai commencé a apprendre comment bien coder, bien penser, bien m'adapter. J'ai eu un aperçu d'un travail plus professionel, et des problèmes auxquels on est confronté. Ma carrière n'a pour l'instant qu'un but : être un bon developpeur, sur qui on peut compter pour apporter de la valeur ajoutée a un projet. Et ce stage m'a montré ce qu'il fallait faire pour atteindre ce but. Merci.

Je déteste le travail. Ca me met la boule au ventre de penser à cette facette de la vie qui force l'individu a consacrer 10 heures par jour a la préparation, déplacement, répétition, et meme sustentation a une entreprise/activité qui tue le psyché a petit feu, pendant des décennies.

Pendant ce stage, j'ai pu écouter de la musique, arreter le travail et rigoler 20 minutes avec un collègue, travailler sur des challenges qui stimulent l'esprit ... et j'ai aimé travailler ici. Puis-je penser qu'un bon environnement est un droit inconditionnel du travail pour lequel se battre ? Et qu'avec , moi aussi, je peut sereinement tenir un travail pendant des decennies ?

Cet espoir m'est **précieux.**