

Documentation de Gestion de Projet

Projet EcoRide - Plateforme de Covoiturage

Le déroulement de mon projet EcoRide

Quand j'ai découvert l'énoncé du projet EcoRide avec ses 13 User Stories, je me suis d'abord senti un peu perdu face à l'ampleur de la tâche. Plutôt que de me lancer directement dans le code, j'ai pris le temps de bien analyser ce qui était demandé. La première chose qui m'a frappé, c'est que le MCD fourni en annexe me semblait incomplet pour répondre à tous les besoins exprimés dans les User Stories.

Partie MCD

J'ai donc passé deux jours entiers à reprendre ce modèle de données depuis le début. Je suis retourné dans chaque User Story, j'ai imaginé concrètement comment les utilisateurs allaient interagir avec l'application, et j'ai reconstruit un MCD plus stable et plus complet. Cette phase d'analyse m'a semblé cruciale, même si j'avais hâte de commencer à coder. Je me disais que mieux valait prendre le temps maintenant que de devoir tout refaire plus tard.

Partie FIGMA

Une fois satisfait de mon modèle de données, je me suis attaqué au maquettage avec Figma, une semaine complète. Au début, je pensais faire quelque chose de rapide, mais je me suis pris au jeu. J'ai voulu créer des maquettes vraiment détaillées, avec des versions desktop et mobile complètement responsives. Je réalise maintenant que j'ai peut-être trop poussé le détail, surtout que le livrable final était un PDF. Mais sur le moment, avoir une vision claire et précise de l'interface finale me rassurait énormément pour la suite du développement.

Partie Développement

Une fois mes maquettes terminées, j'ai pu commencer la phase de développement proprement dite. J'avais la chance d'avoir déjà XAMPP installé et configuré grâce aux cours précédents, donc j'ai pu initialiser rapidement un nouveau projet Git et me lancer. Mon approche a été de développer de manière chronologique, en suivant le parcours utilisateur.

J'ai commencé par créer la structure de base avec `index.php`, en intégrant le header et le footer qui allaient servir sur toutes les pages. Ensuite, je me suis attaqué à la page de recherche, `recherche.php`, qui était centrale dans l'application. Une fois que j'avais une base fonctionnelle pour chercher des covoiturages, j'ai développé tout ce qui tournait autour de la gestion des covoiturages : création, affichage, détails.

La suite logique était de s'occuper de la partie "Mon compte" pour que les utilisateurs puissent gérer leur profil, puis du module de gestion des voitures. C'est à ce moment-là que j'ai réalisé que mon code commençait à devenir un peu brouillon. J'avais tendance à créer un fichier PHP pour chaque formulaire et chaque action, ce qui rendait la navigation dans le code de plus en plus compliquée.

J'ai donc pris la décision de faire une pause dans l'ajout de nouvelles fonctionnalités pour restructurer complètement mon code. J'ai tout reorganisé en classes PHP propres, ce qui a pris moins d'une semaine mais qui a considérablement amélioré la lisibilité et la maintenabilité. J'ai aussi profité de cette phase pour faire des tests de stabilité sur tout ce qui était déjà développé.

Après cette restructuration, j'ai pu reprendre sereinement le développement des fonctionnalités manquantes. J'ai intégré le système d'avis et de notation, puis je suis retourné sur la partie recherche parce que j'avais des problèmes avec la récupération des paramètres de recherche entre les pages. Une fois ce bug résolu, j'ai finalisé avec les filtres avancés et les espaces employé et administrateur.

Pendant ces quatre semaines de développement, j'ai adopté une routine assez stricte. Je me concentrais uniquement sur le PHP et la logique métier, en mettant de côté tout l'aspect visuel. Chaque fois que je bloquais sur quelque chose, je retournais dans les cours ou je cherchais sur internet pour comprendre mes erreurs. J'avais aussi pris l'habitude de valider régulièrement ma logique métier avec des proches qui pouvaient me dire si les règles de gestion me semblaient cohérentes.

Tous les soirs, sans exception, je prenais quelques minutes pour noter où j'en étais. Je mettais des commentaires directement dans le code pour marquer ce qui était terminé, ce qui était en cours, et ce qui devait être debuggé le lendemain. J'avais aussi un carnet à côté de moi où je notais les points importants et les questions à résoudre. Cette documentation au fil de l'eau s'est révélée précieuse, surtout quand je revenais sur du code écrit quelques jours plus tôt.

Partie CSS + Déploiement

Une fois que toute la logique était en place et stable, j'ai consacré ma dernière semaine à l'intégration CSS et au déploiement. J'ai appliqué un style simple mais efficace, en restant fidèle aux maquettes que j'avais créées au début. La phase de

déploiement sur Heroku a été un peu plus compliquée que prévu, avec quelques problèmes de configuration à résoudre, mais j'ai fini par y arriver.