## **Journal de bord du stage**

### **17 mars 2025 - Préparation du projet (avant le codage)**

**Tâches réalisées :**

* Définition du périmètre du projet, du workflow et des exigences fonctionnelles.
* Conception du schéma de base de données (tables : annonces, media, catégories).
* Mise en place de la structure du projet (répertoires, organisation des fichiers).
* Planification des interactions entre les contrôleurs, modèles et méthodes.
* Confirmation du traitement des fichiers médias (stockage temporaire avant soumission).
* Discussion sur les règles de validation et le parcours utilisateur.

**Interactions :**

* Participation à une réunion de présentation d'équipe (9h à 11h) où j'ai rencontré l'équipe.
* Réception de la description détaillée du projet par Guillaume.
* Pose de nombreuses questions à Guillaume pour clarifier certains aspects du projet.
* Prise de notes détaillées publiées sur ma page Notion : [Projet Integration II](https://www.notion.so/Projet-Integration-II-1b9049a845d4806d85a2c89418cf94d0?pvs=4)

**Problèmes rencontrés :**

* Aucun problème rencontré.

### **18 mars 2025 - Implémentation technique**

**Tâches réalisées :**

* Mise en place de la connexion à la base de données (db.php) et validation des requêtes.
* Création des fonctionnalités CRUD pour :
  + Table annonces (gestion des annonces).
  + Table media (liée aux annonces).
  + Table catégories (gestion des catégories).
* Développement du modèle générique (Model.php) comme classe parente pour la gestion des requêtes SQL.
* Implémentation des modèles spécifiques : annonce.php, media.php et categories.php.
* Tests complets des opérations CRUD sur chaque entité (insertion, récupération, mise à jour, suppression).
* Vérification approfondie de l'intégrité relationnelle entre les médias et les annonces.
* Finalisation de la structure globale du projet (contrôleurs, modèles et vues).
* Début de la préparation pour l'intégration de l'interface utilisateur (affichage des annonces et soumission des formulaires).

**Interactions :**

* Début de l’implémentation du backend (connexion base de données, CRUD).
* Échanges ponctuels avec Guillaume pour des clarifications techniques et fonctionnelles sur certaines caractéristiques du projet.
* Début des tests préliminaires sur le frontend en fin de journée.

**Problèmes rencontrés :**

* Quelques difficultés à comprendre clairement certains détails des fonctionnalités demandées.

**19 mars 2025 - Correction de bugs et validation des formulaires**

**Tâches réalisées :**

* Résolution d'un problème lié à Mustache causé par une balise </head> manquante.
* Débogage de l'affichage des catégories dans le formulaire de soumission.
* Correction d'un problème empêchant l'insertion des annonces dans la base de données.
* Ajout de la validation des numéros de téléphone au format international avec intl-tel-input.
* Implémentation d'un système permettant d'afficher/masquer un exemple de numéro de téléphone pour garder le formulaire propre.
* Ajout de logs pour suivre l'exécution des requêtes SQL et déboguer les erreurs d'insertion.
* Tests et ajustements sur l'interface utilisateur pour améliorer l'affichage du formulaire.

**Interactions :**

* Aucune interaction significative avec l'équipe, à part quelques échanges légers.
* La majorité de l'équipe travaillait à distance tandis que je travaillais sur place.

**Problèmes rencontrés :**

* Bug lié à Mustache empêchant l'affichage des catégories, résolu après avoir identifié l'erreur de syntaxe HTML.
* Requêtes d'insertion qui échouaient silencieusement en raison d'un problème avec le champ telephone, qui nécessitait un format numérique valide.
* Complexité de la gestion des numéros de téléphone avec validation internationale.
* Difficulté à maintenir la propreté du formulaire tout en affichant les exemples de formats requis.

**20 mars 2025 - Intégration et validation des formulaires**

**Tâches réalisées :**

* Travail intensif sur l'intégration des formulaires avec Mustache.
* Résolution de plusieurs problèmes liés aux validations de formulaires.
* Ajustements et corrections sur l'affichage des erreurs dans les champs de saisie.
* Débogage des incohérences entre les fichiers Mustache et le système de validation.
* Test et validation des soumissions de formulaires après corrections.
* Vérification de la gestion des erreurs côté client et côté serveur.
* Mise en conformité du formulaire avec les exigences fonctionnelles du projet.
* Ajustement du design et de la responsivité en utilisant Foundation 5.5.1.

**Interactions :**

* Très peu d’interactions avec l’équipe, la majorité travaillant à distance aujourd’hui.
* Discussions légères avec un employé pendant la pause déjeuner.

**Problèmes rencontrés :**

* Plusieurs difficultés avec Mustache pour la validation des formulaires.
* Problèmes d’affichage et de structure du formulaire nécessitant des corrections multiples.
* Bugs liés à l'affichage du menu latéral et du bouton "hamburger" sur mobile.
* Difficulté à comprendre certains comportements de Foundation 5.5.1.
* Problème de surcharge CSS impactant l'affichage du menu.

**Conclusion :**

* Après plusieurs heures de corrections et d'ajustements, les formulaires sont désormais fonctionnels et validés.
* Le design est amélioré, mais quelques ajustements peuvent encore être nécessaires.
* Il reste à tester davantage les interactions utilisateur pour assurer une expérience fluide.

**24 mars 2025 – Réunion d'équipe et travail sur Foundation 5.5.1**

**Tâches réalisées :**

* Réunion avec l’équipe : présentation de l’avancement, discussion sur les objectifs de la semaine prochaine et clarification des tâches à venir.
* Explication de l’implémentation prévue pour la page d’annonces destinée à afficher toutes les annonces.
* Travail spécifique sur l’intégration du framework Foundation 5.5.1 pour la mise en œuvre du design frontend : création d’un header fixe, d’une sidebar positionnée sous le header (affichée sur desktop) et d’un menu hamburger pour mobile.
* Tests et ajustements CSS afin de résoudre les problèmes de positionnement et d’affichage sur desktop.

**Interactions :**

* Échanges constructifs lors de la réunion d’équipe, avec un focus sur la planification des développements à venir.

**Problèmes rencontrés :**

* Quelques difficultés avec le positionnement de la sidebar dues aux règles de position: fixed, résolues par l’ajustement du top (correspondant à la hauteur du header) et de la gestion des marges sur l’inner-wrap.

**Conclusion :**  
La partie design avec Foundation 5.5.1 est désormais fonctionnelle avec un header fixe et une sidebar correctement positionnée sous le header sur desktop. La prochaine étape consistera à implémenter la page d’annonces pour afficher toutes les annonces.

**25 mars 2025 – Amélioration du design, recherche géographique et gestion des filtres**

**Tâches réalisées :**

* Résolution d’un problème critique où les filtres de tri (par ville, par titre, par date) redirigeaient vers la mauvaise route.
* Finalisation de l’affichage des annonces avec image miniature, icône de fichier, et filtrage dynamique côté serveur selon les paramètres de l’URL (type, tri, recherche).
* Amélioration de l’interface utilisateur inspirée de Facebook Marketplace : intégration d’un champ de recherche de ville avec rayon, dans une approche similaire à celle du Marketplace de Facebook.
* Ajout d’un filtre ville + rayon simulé, avec possibilité future de tri géographique précis à l’aide de coordonnées.
* Ajout des colonnes latitude et longitude dans la base de données pour préparer l’intégration géographique.
* Intégration de Geoapify pour l’autocomplétion des villes et récupération automatique des coordonnées GPS.
* Résolution d’un bug où les champs latitude et longitude restaient vides si l’utilisateur tapait une ville sans cliquer sur la suggestion.
* Ajout d’une validation serveur pour empêcher la soumission d’annonces sans coordonnées géographiques valides.
* Débogage approfondi via error\_log() et inspection du payload $\_POST, correction de champs manquants.

**Interactions :**

* Discussion en fin de journée avec le team lead concernant l’approche géographique : décision que les annonces doivent rester limitées à la région de Lanaudière.
* Suite à cette discussion, remise en question de l’utilisation d’une API géographique payante : Geoapify ne couvre pas tous les noms de ville québécois (ex : Notre-Dame-des-Prairies), ce qui remet en cause son utilité.
* Proposition retenue : continuer d’offrir une expérience enrichie avec autocomplétion, mais permettre aussi une validation manuelle si la ville n’est pas détectée.

**Problèmes rencontrés :**

* Bug de formulaire dû à la duplication des champs latitude/longitude (présents deux fois dans le HTML).
* Aucune validation côté client si l'utilisateur entre une ville manuellement sans cliquer dans la liste des suggestions.
* Limitations majeures de la base de données Geoapify : certains noms de ville valides ne sont pas reconnus ou suggérés.
* Difficulté à faire cohabiter validation stricte et flexibilité pour l’utilisateur final.

**26 mars 2025 – Suppression de Geoapify, refonte complète du filtrage géographique**

**Tâches réalisées :**

* Refonte complète du système de filtrage par ville et rayon.
* Suppression de Geoapify suite à la confirmation que tous les utilisateurs sont basés dans Lanaudière (Québec).
* Création d’une nouvelle table localites contenant toutes les villes, municipalités, paroisses et villages de Lanaudière avec leurs latitudes, longitudes, MRC et préfixes postaux.
* Recherche manuelle exhaustive pendant 2 heures pour collecter les codes postaux et la classification municipale pour les 70 localités.
* Mise à jour du formulaire soumission\_offre pour utiliser la localiteId, et auto-remplir ville, mrc et codePostal via JavaScript.
* Modification du contrôleur annonceController::createAnnonce() pour récupérer les données de la localité et éviter toute dépendance API.
* Refonte de getAnnoncesByType() pour filtrer dynamiquement les annonces selon la ville choisie et un rayon en kilomètres (Haversine).
* Ajout d’un affichage dans l’interface utilisateur qui rappelle la ville choisie et le rayon appliqué.
* Refonte complète des pages liste\_offres.html et liste\_besoins.html avec un nouveau design responsive inspiré du Marketplace Facebook.
* Ajout d’un système pour afficher les cartes d’annonces par rangées de 4, avec ajustement CSS.
* Adaptation de la page soumission\_besoin.html pour intégrer les mêmes règles de localité que soumission\_offre.html.

**Interactions :**

* Aucune interaction avec l’équipe aujourd’hui.
* J’étais seul sur place et agissais comme référent technique de la journée.

**Problèmes rencontrés :**

* J’ai perdu beaucoup de temps à comprendre la structure municipale du Québec : il existe plusieurs types d’entités (ville, paroisse, municipalité, village) regroupées en MRC.
* Difficulté initiale à récupérer les préfixes postaux exacts pour chaque localité.
* Conflits de validation dans les formulaires dus à des noms de champs qui ne correspondaient plus après la refonte.
* Problème critique corrigé : codePostal était NULL car le champ caché n’était pas correctement peuplé avant envoi.
* J’ai dû tout tester manuellement après chaque modification pour m’assurer que la validation, le filtrage et les requêtes SQL fonctionnaient comme prévu.

**Conclusion :**  
La transition hors de Geoapify est maintenant terminée. Le système de géolocalisation est 100% local, performant et stable. Toutes les fonctionnalités de recherche, soumission et affichage d’annonces sont compatibles avec cette nouvelle architecture. Le projet est désormais aligné avec les exigences spécifiques du client (limitation à Lanaudière).

**27 mars 2025 – Upload d’images avec compression, galerie et miniature**

**Tâches réalisées :**

* Implémentation d’un **plugin JavaScript réutilisable** pour le téléversement d’images :
  + Sélection d’images avec aperçu instantané (jusqu’à 4 fichiers).
  + Redimensionnement automatique des images à un maximum de 1200x1200 pixels.
  + Compression automatique si la taille dépasse 20MB.
* **Intégration de CropperJS** pour permettre à l’utilisateur de sélectionner une image comme miniature :
  + Affichage d’un bouton “📸 Miniature” sous chaque image.
  + Affichage d’un modal Reveal (Foundation 5.5.1) contenant l’outil de recadrage.
  + Application d’un crop en 1:1 avec surlignage visuel de l’image sélectionnée comme miniature.
* **Ajout du champ d’upload** dans le formulaire de soumission d’une offre.
* **Sauvegarde des fichiers uploadés dans /public/uploads** avec :
  + Génération d’un nom unique.
  + Sauvegarde en base de données (media) du chemin, type MIME, URL publique, et flag is\_thumbnail.
  + Ajout dynamique de la clé url dans le tableau media pour compatibilité avec Mustache.
* **Affichage de la miniature** dans la page liste\_offres via la propriété thumbnail.
* **Affichage d’une galerie d’images** dans la page de détail d’une annonce avec prise en charge conditionnelle :
  + Image / Vidéo / PDF.
  + Fallback visuel si aucune image n’est présente.
* **Correction d’un bug bloquant** lié à l’appel à str\_starts\_with() (incompatible avec PHP < 8.0) remplacé par strpos($type, 'image/') === 0.

**Problèmes rencontrés :**

* Le **modal Reveal de Foundation 5.5.1** ne s’ouvrait pas correctement :
  + Parfois affiché au chargement, parfois impossible à ouvrir.
  + Le comportement bloquant (vraie popup) n’est pas encore totalement fonctionnel.
  + Tentatives avec data-reveal-id, aria-hidden, script JS et triggers foundation('open') sans succès complet.
* L’image recadrée avec CropperJS **ne remplaçait pas l’originale** pour le champ de miniature :
  + Nécessité de corriger la logique de substitution de fichier dans le champ <input type="file">.

**Interactions avec les membres de l’équipe :**

* Discussion avec le **responsable de stage** pour clarifier le fonctionnement attendu du **modal Reveal de Foundation 5.5.1**.
* Il m’a aidé à comprendre les particularités de l’implémentation modale avec Foundation (attributs requis, limitations connues, etc.), ce qui m’a permis d’avancer malgré les difficultés rencontrées.