Raphaël Drecourt
Sofiane Nisar
Enzo Derrien

MI5_D

Rapport Projet Fishing Travel

Etant donné de la bonne cohésion d'équipe que nous avons pu expérimenter au projet du semestre 1, nous avons choisi de garder la même équipe pour ce projet. Cette agence de voyage que nous avons intitulé Fishing Travel, est une agence qui offre différents voyages uniques autour de l'univers de la pêche. Si vous êtes passionné par la pêche alors vous ne pourrez qu'être séduit par la diversité d'offres proposées par Fishing Travel

En ce qui concerne l'organisation de l'équipe, un groupe Discord a été créé ainsi qu'un dépôt Github afin de travailler dans de bonnes conditions notamment pour pouvoir coder tout en communiquant en même temps entre partenaires par le biais d'appels vocaux par exemple.

Répartition des tâches:

- Page d'accueil (Raphaël/Enzo)
- Page de connexion, mot de passe oublié, recherche avancée (Raphaël)
- Page administrateur (Sofiane)
- Page de modification de profil (Sofiane)
- Recherches parallèles [Création des voyages, offres, logements] (Enzo)
- Rapport et Charte graphique (Enzo)
- Page détails voyages/options (Raphaël)
- Page Login/Register (Raphaël)
- Page de réservation (Raphaël)
- Page d'accueil(Raphaël/Sofiane)
- Système de paiement (Sofiane)

- Page recherche voyage (Sofiane)
- Récupération du prix (Raphaël)
- Transactions (Sofiane)
- Fichiers Json (Enzo/Raphaël)
- Création des étapes/options (Enzo)
- Importations d’images (Enzo)
- Thème clair/sombre (Raphaël)
- Connexion/Inscription (Raphaël)
- Profil (Sofiane)
- Admin (Sofiane)
- Panier (Sofiane)
- Prix temps réel recap (Sofiane)
- Vérification de la date de début d'un voyage (Sofiane)
- Mise à jour données utilisateur via le profil utilisateur (Sofiane)
- Mise à jour données utilisateur via la page de contrôle administrateur (Sofiane)
- Javascript pour la page de modification d'un utilisateur via la page de contrôle administrateur (Sofiane)
- Actualisation du prix grâce à une requête vers le serveur (Raphaël)

Planning de la réalisation:

Phase 1:

Semaine 1:

- Découverte du Projet
- Réalisation de la page d'accueil {Aide du site Figma}
- Création de la page de connexion

Semaine 2:

- Page Administrateur
- Page Modification de profil
- Page Mot de passe oublié
- Page de Recherche
- Créations de voyages
- Rapport (Enzo)

Difficultés rencontrées:

Pour le moment, peu de difficultés particulières ont été rencontrées, si ce n'est pour ma part (Enzo) dans la réalisation du footer où l'alignement des éléments ne me convenaient pas malgré l'utilisation de text-align dans le CSS. Mais en ajoutant du padding 15px ce léger problème fut résolu.

Phase 2 :

Semaine 1:

- Page détails voyages/options
- Page Login/Register
- Page de réservation
- Page d'accueil
- Fichiers Json
- Création des étapes/options
- Boutons page administrateur

Semaine 2 :

- Système de paiement
- Page recherche voyage
- Récupération du prix voyage
- Transactions
- Importations d'images
- Rédaction du rapport

Difficultés rencontrées :

En phase 2, nous avons dû effectuer à nouveau du code CSS pour améliorer certains détails ce qui nous a pris du temps. On a également rencontré des difficultés au niveau du système de paiement et dans la recherche des voyages où le fichier JSON a été relativement compliqué à parcourir et à déterminer si chaque entrée était pertinente par rapport à la recherche. La pagination de la page de recherche nous a également posé quelques soucis car il fallait changer de page tout en gardant les données de la recherche dans le GET de l'URL.

Architecture :

Les noms des JSONs et leurs attributs sont explicites concernant leurs utilités.

- **reservations.json :** liste toutes les réservations effectuées sur le site

ID de la réservation, ID utilisateur, ID du voyage réservé, date de la réservation, nombre de personnes, dictionnaire des options sélectionnées avec leur quantité.

- **transactions.json :** liste toutes les transactions effectuées sur le site
  n° de la transaction (string), date, email du compte utilisé pour la réservation, l'ID de la réservation associée (≠ transaction), montant.
  "_comment" est utilisé (pour la première entrée uniquement) pour le fonctionnement du code servant à attribuer un numéro de transaction (string) à chaque transaction. Cette décision à été prise car la fonction *count()* retourne NULL si un fichier est vide (= s'il n'y a pas de transactions dans le JSON), entraînant des erreurs.
- **users.json :** liste des utilisateurs inscrits sur le site
  ID utilisateur, pseudo, email, mot de passe haché, rôle (client, vip, admin)
- **voyages.json :** liste tous les voyages, étapes et les options associées
  ID voyage, titre voyage, description (utilisé sur la page du voyage), prix, durée, pays, étapes ("sous-JSON" : secteur du voyage, poissons à pêcher, hôtels (nom, adresse), durée étape, options (nom option, type (individuel ou groupe), prix option, description))

Les images utilisées pour chaque voyage sont stockées dans *"assets/voyages/ID"* avec ID l'ID du voyage en question.
Les scripts (fichiers PHP qui ne génèrent pas de code HTML) sont situés dans *"script"*.
Les fonctions utilisées fréquemment sont situées dans *"include"*.
À noter que plusieurs transactions peuvent pointer vers la même réservation.

Mot de passe Admin1, Admin2 : admin
Mot de passe User1, User2, User3 : 1234

Phase 3 :

Semaine 1 :

- Thème clair/sombre
    - Nous avons remplacé le cookie par un stockage interne en JavaScript (localStorage). Le fonctionnement est équivalent.
- Connexion/Inscription
- Boutons dynamiques du profil
    - Étant donné que nous mettons à jour les données utilisateurs directement en PHP, nous attendons 5 secondes avant de soumettre le formulaire (comme indiqué dans les consignes de la phase 3).

Semaine 2 :

- Page administrateur, avec boutons dynamiques
- Panier (server side)
- Actualisation des prix en temps réel sur la page récapitulative du voyage sélectionné
- Vérification de la date de début d'un voyage (il faut que le voyage commence au minimum le lendemain de la date d'aujourd'hui)

Difficultés rencontrées :

Durant la phase 3, le Javascript empêchait la transmission des données du formulaire en PHP. Pour y remédier, nous avons donc recréé les champs manuellement en JavaScript, puis nous avons soumis le formulaire manuellement avec form.submit(). Ensuite, il était compliqué de transmettre des variables PHP que nous voulions utiliser dans le JavaScript. Nous avons donc utilisé "data-*variable*" dans les balises HTML en injectant des variables PHP dedans pour pouvoir faire le transfert des variables PHP vers le JavaScript. (document.body.dataset.*variable*)

Phase 4 :

- Mise à jour données utilisateur via profil
- Mise à jour données utilisateur via la page de contrôle administrateur (données utilisateur, vip, bannissement)
- Javascript pour la page de modification d'un utilisateur via la page de contrôle utilisateur (rien à voir avec la phase 4 sauf la requête ASYNC, on avait oublié de mettre cette page à jour pour la phase 3)
- Correction d'un bug : dans la page du profil utilisateur, le bouton de soumission apparaissait après validation même lorsque aucun champ n'avait été modifié (rien à voir avec la phase 4)

Particularités:

- Nous avons fait en sorte d'actualiser le prix grâce à une requête vers le serveur sans refresh la page et non en front-end comme c'était auparavant
- Choix d'utiliser l'API Fetch (syntaxe async / await) car Jquery et AJAX sont trop vieux (d’après les cours magistraux)