

# Rapport Projet informatique 4

## CyCruise



Antonin PERALTA  
Alexandre Halhoute

Sujet ClickJourneyY  
PréIng-2 Math-Info options Physiques  
MI-1 Tri-J  
Année 2024/2025

# Sommaire

## a) PHASE 1 (HTML & CSS)

- i) Répartition des tâches
- ii) Difficultés rencontrées

## b) PHASE 2 (PHP)

- i) Répartition des tâches
- ii) Difficultés rencontrées

## c) PHASE 3 (JavaScript + DOM)

- i) Répartition des tâches
- ii) Difficultés rencontrées

# PHASE - 1

## 1) Répartition des tâches

Nous avons réparti les tâches en fonction du niveau et des compétences de chacun en développement HTML / CSS. Nous avons donc répartie les tâches de travail de cette manière.

Antonin : Étant plus à l'aise avec le HTML et le CSS, il a d'abord travaillé sur la structure principale du projet en créant les fichiers CSS “app.css” et “btn-kit.css” Le premier permet de centraliser les couleurs du projet grâce au système de variables CSS, tout en assurant une base HTML propre en supprimant les marges par défaut et en définissant la police d'écriture. Le fichier btn-kit.css, quant à lui, facilite l'utilisation de différents types de boutons sans avoir à les recréer dans chaque fichier CSS. Ensuite, il s'est chargé du système de navigation (navbar) et du footer des page, en réalisant le code HTML ainsi que les fichiers CSS nécessaires, afin de les réutiliser sur toutes les pages. Enfin, il a développé la plupart des pages accessibles au public : home, cruiselist, login/register, account et about.

Alexandre : En découvrant le HTML ainsi que le CSS, Alexandre a rencontré des difficultés pour le projet. Malgré cela, il a pris en charge la création des pages d'administration. Malgré les obstacles, il a réussi à concevoir un template avec une barre d'informations en haut ainsi qu'une barre de navigation sur la gauche (à l'instar du panneau d'administration de WordPress, pris en exemple). Après avoir finalisé ce template avec l'aide d'Antonin, il a pu créer la page d'administration par défaut ainsi que la liste des utilisateurs.

Nous avons essayé d'exploiter au mieux les aptitudes de chaque membre du groupe, mais aussi de rendre meilleur les personnes étant plus en difficultés, afin d'avoir le projet le plus propre possible.

## 2) Difficultés rencontrées

Une des premières difficultés que nous avons rencontrées était d'avoir un projet bien structuré. En effet, comme c'était l'une des premières fois que nous devions créer un site web, nous avons eu des difficultés à définir une architecture de projet claire. Nous avons donc décidé de placer toutes les pages HTML à la racine du projet, sauf les pages dites "admin", qui se trouvent dans un sous-dossier admin. Pour tout ce qui concerne les fichiers externes (CSS, images, polices), nous les avons regroupés dans un dossier "public".

Sur la page de compte, nous avons rencontré des difficultés à aligner correctement les trois colonnes sur la même ligne tout en les rendant lisibles. De plus, nous souhaitions ajouter un "scrollbar" pour chaque colonne. Cela a été compliqué, mais nous avons réussi à le résoudre grâce aux attributs CSS max-height et overflow-y.

Sur la page croisière-détail (qui, pour le moment, affiche les détails d'une seule croisière), nous avons rencontré des difficultés sur la partie des réservations. En effet, cette section est constituée de plusieurs colonnes représentant des options, et chaque ligne contient un bouton correspondant à une option disponible. Il a donc fallu travailler avec les forms afin de proposer plusieurs options pour une même catégorie, tout en s'assurant qu'une option se décoche automatiquement lorsqu'une autre est sélectionnée. Pour cela, nous nous sommes tournés vers les formulaires de type "radio". Cependant, nous avons rencontré des difficultés pour styliser les boutons et rendre bien visible la différence entre une option cochée et une option non cochée. Finalement, grâce au sélecteur CSS ":checked + label", nous avons réussi à obtenir le résultat souhaité.

Pour la page "About", nous avons voulu intégrer une vidéo en première page tout en conservant la barre de navigation et en ajoutant du texte par-dessus. Au début, nous avons rencontré de nombreuses difficultés pour que la vidéo occupe toute la page et passe correctement derrière la navbar. Cependant, nous avons réussi à trouver une solution grâce au système de gestion des calques (z-index).

Pour finir avec les pages d'administration, nous avons voulu mettre en place un système de sidebar (c'est-à-dire une navbar fixée sur la gauche plutôt que sur le haut de la page) tout en conservant une barre d'informations sur la gauche. Cela a été relativement complexe, car il a fallu travailler avec les flexbox tout en veillant à ce que toutes les pages s'alignent correctement sur la marge de la navbar, faute de quoi elles risquent de se superposer.

# PHASE - 2

## 1) Répartition des tâches

Nous avons réparti les tâches en fonction du niveau et des compétences de chacun en développement PHP et MariaDB. Voici la répartition du travail :

**Antonin** : Étant plus à l'aise avec PHP et MariaDB, il a d'abord travaillé sur la structure principale du projet en créant les schémas SQL ainsi que l'organisation des dossiers ("public", "src", "component"). Il a également implémenté la plupart des fonctionnalités, notamment l'authentification des utilisateurs et la gestion des croisières. Avec cela on a donc l'implémentation des système dans les page publique `account.php` / `cruise-list` etc

**Alexandre** : En découvrant PHP, Alexandre a rencontré des difficultés au cours du projet. Malgré cela, il a pris en charge certaines pages d'administration. Il a réussi à concevoir une page affichant la liste des utilisateurs, qu'il a ensuite pu réutiliser pour d'autres pages similaires (liste des croisières, liste des bateaux, etc.). Avec l'aide d'Antonin, il a également créé la page d'administration permettant d'ajouter une croisière et de modifier un utilisateur. Il a aussi le html/css de certaine page qui n'était pas encore faite (ex `payment-callback`)

Nous avons cherché à exploiter au mieux les compétences de chaque membre du groupe tout en aidant ceux rencontrant plus de difficultés, afin d'obtenir un projet le plus propre et abouti possible.

## 2) Difficultés rencontrées

L'un des principaux problèmes que nous avons rencontrée est en rapport avec le système de données utilisé mariadb, notamment sur la gestion des croisières avec les multiples relations pour les options et les étapes. Pour résoudre ce problème nous avons simplement connecté les class de type repository entre elles c'est à dire que lorsque l'ont crée l'objet croisière nous allons utilisé les 3 class repository la principal celle pour les options et pour finir celle des stages.

Un autre problème rencontré concernait la page de réservation, notamment pour le calcul du prix et l'ajout de personnes à la croisière. Étant donné que nous ne pouvions pas utiliser JavaScript, il a été difficile de mettre en place un système propre pour gérer les options de réservation.

Pour pallier ce problème, nous avons opté pour un bouton "**Mettre à jour**", qui envoie le formulaire à la page et modifie ainsi les champs en conséquence. Cependant, sans le savoir, cela nous a également posé un problème avec le bouton "**Créer une réservation**", car nous devions envoyer le même formulaire vers deux pages PHP différentes. Nous avons finalement trouvé la solution en utilisant l'attribut **formaction** sur le bouton de soumission du formulaire.

Le dernier problème était lié à l'accès à notre projet, car nous utilisons une base de données MariaDB. En effet, l'utilisation de ce service rend plus complexe le lancement de notre application PHP. Pour résoudre ce problème, nous nous sommes tournés vers la technologie **Docker**, en créant une image Docker adaptée à notre projet, qui installe automatiquement toutes les dépendances nécessaires pour **MariaDB** et **MySQL**. Une fois cette image créée, nous avons mis en place un fichier **Docker Compose** nommé "**dev-compose.yml**", permettant de créer automatiquement un serveur MariaDB et de le connecter à l'application **CyCruise** du projet. Grâce à cette solution, en seulement deux commandes, le service est lancé et fonctionne parfaitement.

**NOTE IMPORTANTE** : TOUTES LES INFORMATIONS TECHNIQUES IMPORTANTES SE TROUVENT DANS LE DOCUMENT "**DOSSIER TECHNIQUE**" NOMMÉ "**DT**".

#### LISTE DES COMPTE PAR DEFAULT

email: test.admin@localhost password: password role: admin  
email: test1.admin@localhost password: password role: admin  
email: test.user@localhost password: password role: user  
email: test.vip@localhost password: password role: vip  
email: test.premium@localhost password: password role: premium

# PHASE - 3

## 1) Répartition des tâches

Pour cette phase qui été plus légère que la dernière il y'a eu moins de tâche à effectuer, donc la répartition des tâches a été plus facile

**Antonin:** Il a fait principalement la partie des triés sur la page cruise-list ainsi que l'auto update des prix sur la page de réservation, et pour finir le dark theme car c'est lui qui avait fait les fichiers CSS des couleurs

**Alexandre :** Alexandre a fait les fichiers javascript plus simple tels que la vérification du login & du register, aussi il a tenté de faire le système de délayé pour la page admin mais a finalement du être aidé par antonin.

## 2) Difficultés rencontrées

Le principal problème que nous avons rencontré est que nous ne pouvons pas utiliser fetch, il a donc fallu trouver un moyen pour récupérer les données depuis le fichier HTML. Pour cela, nous nous sommes tournés vers le système de "data" de HTML & DOM qui nous permet d'écrire dans notre fichier HTML "data-coucou" et de pouvoir le reprendre dans le JS via dom grâce à "element.dataset["coucou"]" Un autre souci que nous avons eu est l'ordre de chargement des fichiers JS et CSS pour le dark theme, car il était possible que celui-ci ne s'affiche pas en fonction de l'ordre de chargement.