

# Rapport de Projet

## Informatique 4

Présenté par : KONDA-MOUGNONGUI Jérémie  
BOUCHAM Jibril  
KAE-NUNE Damien

Dirigé par : LE BRETON Caryl

16 mai 2025

## Table des matières

<b>Introduction</b>	<b>3</b>
<b>Planning</b>	<b>3</b>
Phase #1 . . . . .	3
Phase #2 . . . . .	3
Phase #3 . . . . .	3
Phase #4 . . . . .	3
<b>Répartition des Tâches</b>	<b>4</b>
Phase #1 . . . . .	4
Phase #2 . . . . .	4
Phase #3 . . . . .	4
Phase #4 . . . . .	5
<b>Fonctionnalités Implémentées</b>	<b>5</b>
Phase #2 . . . . .	5
Phase #3 . . . . .	6
Phase #4 . . . . .	8
<b>Problèmes Rencontrés et Solutions Apportées</b>	<b>9</b>
Phase #1 . . . . .	9
Phase #2 . . . . .	10
Phase #3 . . . . .	10
Phase #4 . . . . .	11
<b>Conclusion</b>	<b>12</b>
Conclusion de la Phase #1 . . . . .	12
Conclusion de la Phase #2 . . . . .	12
Conclusion de la Phase #3 . . . . .	12
Conclusion de la Phase #4 . . . . .	12
<b>Bilan du projet</b>	<b>13</b>

## Introduction

Ce document constitue le rapport de projet pour les 4 phases. Il présente :

- Le planning réalisé durant ces phase.
- La répartition des tâches entre les membres du groupe.
- Les problèmes humains, organisationnels et techniques rencontrés.
- Les solutions mises en place pour y remédier.

## Planning

### Phase #1

Voici le planning réalisé par le groupe pour la Phase #1 :

- **Semaine 1** : Analyse du cahier des charges et brainstorming. Réalisation de la maquette Figma ainsi que de la charte graphique du projet.
- **Semaine 2** : Répartition des tâches et élaboration du planning pour un rendu prêt pour Samedi. On s'est penché sur l'aspect technique/code, avec la production des différentes pages en HTML/CSS.

### Phase #2

Voici le planning réellement suivi par l'équipe pour cette phase :

- **Semaine 1** : Conception accélérée du format JSON et restructuration des fichiers. Priorisation des structures utilisateurs et voyages.
- **Semaine 2** : Développement partiel des fonctionnalités critiques (inscription, connexion) en parallèle d'un projet externe (webscrapping et analyse de données en Python pour un cours d'économie urbaine).
- **Semaine 3** : Travail intensif le week-end et en soirée pour rattraper le retard : personnalisation des voyages, récapitulatif et ébauche de la page de paiement. Finalisation en urgence des vues (liste/détail des voyages), amélioration CSS et tests partiels. La page de paiement reste minimaliste.

### Phase #3

Voici le planning suivi pour la Phase #3 :

- **Semaine 1** : Revue du cahier des charges de la Phase 3. Répartition des nouvelles tâches JavaScript. Intégration de Damien de retour dans l'équipe. Correction des bugs résiduels de la Phase 2 (notamment finalisation du paiement et robustesse des sessions). Mise en place des branches Git par fonctionnalité/membre.
- **Semaine 2** : Développement intensif des fonctionnalités JavaScript : changement de thème, validation des formulaires côté client, interactivité des pages profil et admin, tri des voyages, calcul dynamique du prix. Tests et intégration des différentes branches.

### Phase #4

Voici le planning suivi pour la Phase #4 :

- **Semaine 1** : Revue du cahier des charges de la Phase 4. Répartition des nouvelles tâches. Correction des bugs résiduels de la Phase 3 (notamment finalisation du paiement, robustesse des sessions et du panier). Mise en place d'un tableau des issues sur GitHub pour une meilleure gestion de projet.
- **Semaine 2** : Développement intensif des fonctionnalités requises pour la Phase 4 (mises à jour asynchrones, chargement dynamique), développement d'un "easter egg", préparation des trois scénarios de démonstration (utilisateur, réservation, chaos), rédaction du rapport final et du fichier README, ainsi que la résolution des bugs identifiés.

# Répartition des Tâches

## Phase #1

La répartition des tâches pour la Phase #1 s'est organisée comme suit, Damien étant absent durant cette période :

- **Jéréemie :**
  - Conception de l'interface utilisateur et création des maquettes Figma.
  - Développement HTML/CSS des pages à complexité visuelle simple (ex : page d'accueil, page de présentation "Qui sommes-nous?").
- **Jibril :**
  - Développement front-end principal en HTML/CSS, mettant à profit ses compétences solides dans ces langages.
  - Collaboration avec Jéréemie sur l'intégration HTML/CSS globale du projet.
- **Damien :**
  - Absent durant cette phase.

## Phase #2

Durant la Phase #2, Damien était toujours absent. La répartition s'est donc faite entre Jéréemie et Jibril :

- **Jéréemie :**
  - Développement principal en PHP, étant plus à l'aise avec ce langage.
  - Conception et développement de la page "Destination" de A à Z.
  - Gestion de la logique des voyages et du processus de réservation.
  - Création du fichier JSON `voyages.json` (liste des voyages et leurs détails).
  - Création du fichier JSON `paiements.json` (structure initiale).
  - Implémentation de l'API de paiement (avec quelques difficultés initiales).
- **Jibril :**
  - Développement PHP pour les fonctionnalités utilisateur : inscription, connexion, gestion de la page admin (structure initiale).
  - Implémentation des sessions PHP pour la gestion des utilisateurs.
  - Conception et implémentation du système de hachage des mots de passe, en collaboration avec Jéréemie pour l'intégration avec le format JSON des utilisateurs.
- **Damien :**
  - Absent durant cette phase.

## Phase #3

Le retour de Damien a coïncidé avec des contraintes personnelles et professionnelles pour Jéréemie. La répartition a été adaptée en conséquence :

- **Jéréemie :**
  - Implémentation (partielle et perfectible en raison de contraintes de temps et de santé) du tri des voyages côté client.
  - Développement du calcul dynamique des prix en JavaScript sur la page de détail du voyage (logique similaire au PHP, facilitant la prise en main).
  - Rédaction et mise à jour du rapport de projet.
- **Jibril :**
  - Collaboration étroite avec Damien sur l'amélioration des pages "Profil Utilisateur" et "Admin" (aspects JavaScript et CSS).
  - Aide à Damien pour la compréhension et l'intégration CSS du mode clair/sombre.
- **Damien :**
  - Développement complet de la fonctionnalité de changement de thème (mode clair/sombre) avec persistance via cookie, avec l'appui de Jibril pour le CSS.

- Développement de la fonctionnalité "Panier", reprenant le travail initialement prévu par Jérémie.
- Amélioration des interactions JavaScript sur les pages "Profil Utilisateur" et "Admin", en collaboration avec Jibril.

## Phase #4

Pour cette phase finale, axée sur l'asynchronisme et la finalisation, la répartition s'est équilibrée comme suit :

- **Jérémie :**
  - **Fonctionnalités Asynchrones :** Implémentation des requêtes AJAX pour le calcul dynamique du prix total du voyage en fonction des options et pour le chargement dynamique des options de voyage (en collaboration avec Damien).
  - **Finalisation Projet :**
    - Rédaction finale du rapport et du fichier README.md.
    - Correction de bugs majeurs (ex : problèmes d'affichage du header, erreurs dans l'affichage des voyages, dysfonctionnements du calcul des options).
    - Développement de l'"easter egg".
    - Préparation et simulation du scénario de démonstration "Réservation utilisateur" (parcours sans accroc).
    - Contribution à la sécurité des données PHP pour le scénario "Chaos".
- **Jibril :**
  - **Fonctionnalités Asynchrones :** Implémentation des requêtes AJAX pour la mise à jour asynchrone du profil utilisateur et pour la gestion asynchrone dans la vue administrateur (avec l'aide de Damien).
  - **Finalisation Projet :**
    - Préparation et simulation du scénario de démonstration "Utilisateur" (navigation, profil, etc.).
- **Damien :**
  - **Fonctionnalités Asynchrones :** Soutien technique et collaboration sur l'ensemble des quatre fonctionnalités AJAX (celles de Jérémie et celles de Jibril), apportant son expertise en JavaScript et AJAX.
  - **Finalisation Projet :**
    - Préparation et simulation du scénario de démonstration "Chaos" (tests de robustesse, tentatives de contournement), en collaboration avec Jérémie pour les aspects PHP et sécurité.

## Fonctionnalités Implémentées

### Phase #2

#### Format des Données et Organisation des Fichiers

Pour cette phase, nous avons choisi d'utiliser le format JSON pour stocker toutes nos données. Ce choix a été motivé par plusieurs raisons :

- Facilité de manipulation en PHP avec les fonctions `json_encode()` et `json_decode()`
- Structures de données hiérarchiques adaptées à notre modèle
- Lisibilité et maintenabilité du code

Conformément aux exigences, nous avons créé 5 utilisateurs prédéfinis :

- 2 administrateurs (admin1, admin2)
- 3 utilisateurs normaux (utilisateur1, utilisateur2, utilisateur3)

**Organisation des Fichiers de Données :** Nous avons organisé nos fichiers de données dans une structure séparée des scripts PHP :

/data/

```
data_user.json      // Informations sur les utilisateurs
voyages.json        // Liste des voyages disponibles
paiements.json      // Historique des paiements et configurations achetées(Coming Soon...)
```

Conformément aux exigences, nous avons créé 15 voyages différents (dont 3 avec les options de fait ) avec des configurations variées (nombre d'étapes, options, prix, dates...) pour permettre de tester l'ensemble des fonctionnalités du site.

## Système d'Authentification

Nous avons mis en place un système d'authentification complet avec :

- Inscription d'un nouvel utilisateur avec vérification des données
- Connexion par login/mot de passe
- Gestion des sessions PHP pour maintenir l'état de connexion
- Redirection automatique selon l'état de connexion
- Protection des pages nécessitant une authentification

## Affichage des Voyages

Nous avons implémenté deux types d'affichage des voyages :

1. **Affichage en page de destination** : Une sélection de voyages recommandés visible par tous les utilisateurs
2. **Affichage suite à une recherche** : Une liste filtrée selon les mots-clés entrés par l'utilisateur

## Vue Détaillée et Personnalisation des Voyages

Pour la vue détaillée d'un voyage, nous avons créé une page qui :

1. Affiche les informations générales du voyage
2. Liste toutes les étapes avec leurs options par défaut
3. Permet de modifier les options pour chaque étape via des formulaires

Les modifications d'options sont stockées temporairement dans la session utilisateur.

## Récapitulatif et Paiement

Non fonctionnel pour le moment.

## Phase #3

Cette phase s'est concentrée sur l'ajout d'interactivité côté client grâce à JavaScript, sans rechargement de page pour de nombreuses actions :

## Changement de Thème

- Un bouton (ou autre contrôle) permet de basculer entre au moins deux thèmes CSS (ex : clair/sombre).
- Le changement applique une nouvelle classe au `<body>` ou charge un fichier CSS alternatif via JavaScript, modifiant l'apparence sans recharger la page.
- Le choix de l'utilisateur est sauvegardé dans un `cookie` pour être appliqué lors des prochaines visites ou chargements de page.

## Validation des Formulaires Côté Client

- Les formulaires d'inscription et de connexion sont validés en JavaScript avant l'envoi au serveur.
- Des messages d'erreur clairs s'affichent à côté des champs invalides sans recharger la page.
- La soumission du formulaire n'est autorisée que si tous les champs sont valides.
- Ajout d'un bouton "œil" pour afficher/masquer les mots de passe.
- Ajout de compteurs de caractères en temps réel pour les champs avec une longueur limitée (ex : pseudo, mot de passe).

## Page de Profil Utilisateur Interactive

- Les champs du profil sont initialement affichés en lecture seule (ou grisés).
- Un bouton "modifier" à côté de chaque champ (ou groupe de champs) le rend éditable.
- Des boutons "Valider" et "Annuler" apparaissent pour le champ en cours d'édition.
- Cliquer sur "Annuler" restaure la valeur initiale et rend le champ non éditable, sans recharger la page.
- Cliquer sur "Valider" enregistre temporairement la modification (en JS) et rend le champ non éditable, sans recharger la page.
- Un bouton global "Enregistrer les modifications" apparaît si au moins un champ a été validé, permettant d'envoyer l'ensemble des modifications au serveur via un formulaire PHP classique (comme avant).

## Page Administrateur (Simulation d'Update)

- Sur la page listant les utilisateurs, les actions de modification (ex : changer le rôle, bannir via un switch/bouton) déclenchent une simulation visuelle.
- Le contrôle (bouton/slider/checkbox) est grisé/désactivé pendant quelques secondes (simulant un appel réseau).
- Après ce délai, le contrôle est réactivé et affiche la nouvelle valeur (visuellement uniquement en Phase 3). L'envoi réel des données au serveur est prévu pour la Phase 4 (requêtes asynchrones).

## Tri des Résultats de Recherche Côté Client

- Sur la page affichant les résultats d'une recherche de voyages, des options de tri (par date, prix, durée, nombre d'étapes) sont disponibles.
- Cliquer sur une option de tri réorganise la liste des voyages affichés directement en JavaScript, sans recharger la page ni effectuer de nouvelle requête HTTP.

## Calcul Dynamique du Prix Estimé

- Sur la page de détail/personnalisation d'un voyage, la modification d'une option (hébergement, activité, nombre de personnes...) met à jour dynamiquement le prix total estimé affiché sur la page.
- Ce calcul est effectué entièrement en JavaScript, en se basant sur les prix des options (qui doivent être disponibles côté client, par exemple via des attributs `data-*` ou un objet JSON initial).
- La mise à jour se fait sans recharger la page et sans requête HTTP. Le prix final calculé côté serveur (Phase 2) lors du récapitulatif doit rester cohérent avec ce prix estimé.

## Fonctionnalité "Panier" (Concept)

- Mise en place d'une logique pour suivre le(s) voyage(s) qu'un utilisateur connecté est en train de personnaliser.
- Cela peut impliquer de sauvegarder l'ID du voyage et les options choisies en cours de modification dans la session PHP ou potentiellement dans le `localStorage` du navigateur, permettant à l'utilisateur de reprendre plus tard (bien que la persistance réelle via `localStorage` soit plus

avancée). L'objectif principal est d'avoir une référence au voyage actif pour les étapes de récapitulatif et de paiement.

## Organisation du Code JS

- Le code JavaScript a été structuré dans des fichiers séparés, inclus dans les pages HTML/PHP concernées, améliorant la maintenabilité.

## Améliorations Phase Précédente

- Correction de bugs et amélioration de la robustesse des fonctionnalités développées en Phase 2, notamment la gestion des sessions et le processus de paiement.

## Phase #4

Cette phase finale a été consacrée à l'implémentation de requêtes asynchrones pour améliorer significativement la fluidité et la réactivité de l'application, ainsi qu'à la finalisation de plusieurs aspects du projet.

### Mise à Jour Asynchrone du Profil Utilisateur

Lors de la mise à jour des informations personnelles dans le profil utilisateur, les modifications sont désormais gérées de manière asynchrone :

- L'envoi des données modifiées s'effectue via une requête JavaScript (Fetch API) vers un script PHP dédié, sans rechargement de la page.
- Le script PHP traite la demande, met à jour le fichier `data_user.json` et renvoie une réponse (généralement en JSON) indiquant le succès ou l'échec de l'opération.
- Côté client, JavaScript interprète la réponse :
  - En cas de succès, les nouvelles valeurs sont affichées et confirmées à l'utilisateur.
  - En cas d'échec (par exemple, validation côté serveur échouée), un message d'erreur est affiché et les champs sont réinitialisés à leurs valeurs précédentes si nécessaire.
- L'interface utilisateur est mise à jour dynamiquement pour refléter le statut de la modification.

### Gestion Asynchrone dans la Vue Administrateur

Les modifications d'informations utilisateur depuis la vue administrateur (changement de rôle, statut, etc.) sont également opérées via des requêtes asynchrones :

- Une requête JavaScript est envoyée au serveur lorsqu'une action administrative est initiée.
- Un indicateur visuel (par exemple, une icône de chargement ou un "spinner") est affiché à côté de l'élément modifié pour informer l'administrateur que l'opération est en cours.
- Pour simuler la latence réseau et permettre de visualiser clairement cet indicateur lors des démonstrations, un délai artificiel (par exemple, `sleep(1)` ou `sleep(2)`) a été ajouté côté serveur PHP avant le traitement de la requête.
- Après réception de la réponse du serveur (succès/échec), l'indicateur de chargement est masqué et l'interface est mise à jour pour refléter le nouveau statut de l'utilisateur.

### Chargement Dynamique des Options de Voyage

Dans la page de personnalisation d'un voyage, les listes d'options disponibles (par exemple, types d'hébergement, activités) sont désormais chargées dynamiquement :

- Au lieu d'être codées en dur dans le HTML, les listes déroulantes (ou autres sélecteurs d'options) sont initialement vides ou contiennent des options par défaut.
- Lorsque cela est nécessaire (par exemple, au chargement de la section d'une étape ou en fonction d'une sélection précédente), JavaScript effectue une requête asynchrone vers un script PHP.



- Ce script PHP récupère les options pertinentes (par exemple, à partir d'un fichier JSON de configuration des voyages ou d'une logique métier) et les renvoie au client (généralement au format JSON).
- JavaScript peuple ensuite dynamiquement les éléments de formulaire (ex : `<select>`) avec les options reçues, sans recharger la page.

## Calcul Asynchrone du Prix Total du Voyage

Le calcul du prix total d'un voyage, en fonction des options sélectionnées par l'utilisateur, s'effectue de manière asynchrone :

- À chaque modification d'une option impactant le prix (nombre de personnes, choix d'une activité payante, type d'hébergement, etc.), JavaScript collecte l'ensemble des options actuellement sélectionnées.
- Ces informations sont envoyées via une requête asynchrone à un script PHP dédié au calcul du prix.
- Le script PHP applique la logique tarifaire (potentiellement complexe et stockée côté serveur pour des raisons de sécurité et de maintenabilité) et retourne le prix total actualisé.
- JavaScript reçoit cette réponse et met à jour l'affichage du prix total sur la page en temps réel, sans nécessiter de rechargement.

## Améliorations et Finalisation

Outre les fonctionnalités asynchrones principales, cette phase a inclus :

- **Finalisation du Panier** : Amélioration de la logique de gestion du panier, assurant une meilleure persistance des voyages en cours de personnalisation et une transition fluide vers le récapitulatif et le paiement.
- **Résolution de Bugs** : Correction de divers bugs identifiés lors des phases précédentes ou lors des tests intensifs de la Phase 4, améliorant la stabilité et la robustesse globale de l'application.
- **Pages Complémentaires** : Implémentation de pages "hors phase" initialement, telles que les Conditions Générales d'Utilisation (CGU) et une page de contact, pour enrichir le site.
- **Easter Egg** : Ajout d'une fonctionnalité cachée ("easter egg") pour apporter une touche ludique au projet.

# Problèmes Rencontrés et Solutions Apportées

## Phase #1

### Problèmes Humains

- **Problème** : Communication lente entre les membres du groupe lors de la semaine 1.
- **Solution** : Mise en place de réunions après les cours et pendant les pauses pour maximiser notre temps sur le projet.

### Problèmes Organisationnels

- **Problème** : Difficulté à se mettre au travail en avance en rapport avec la deadline.
- **Solution** : Révision du planning et réaffectation des tâches pour une meilleure répartition de la charge de travail.

### Problèmes Techniques

- **Problème** : Intégration des différentes parties du projet dans Git.
- **Solution** : Pour l'instant le commit se fait uniquement par upload, on compte se former pour utiliser Git et faire des commit régulier.

## Phase #2

### Contraintes Humaines

- **Problème** : Charge de travail excessive pour Jérémie (projet d'économie urbaine avec webs-crapping + rapport à rendre jeudi). Délais compromis.
- **Solution** : Travail nocturne et week-end, délégation accrue à Jibril pour les pages critiques (authentification).
- **Problème** : Absence de Damien (prévu pour la Phase 3), équipe réduite à 2 membres.
- **Solution** : Regroupement des tâches de nouvelles pages sous Jérémie et page déjà existante sous Jibril malgré les compétences croisées.

### Défis Organisationnels

- **Problème** : Sous-estimation majeure du temps nécessaire pour :
  - Lier les données JSON aux templates PHP
  - Gérer les dépendances entre options de voyage (ex : hébergement → impact sur le prix total)
- **Solution** : Simplification des JSON et utilisation de valeurs par défaut pour les options complexes.

### Difficultés Techniques

- **Problème** : Incohérences des sessions PHP lors de la navigation entre personnalisation et récapitulatif.
- **Solution** : Ajout d'un système de validation étape par étape et purge des données temporaires.
- **Problème** : Calcul des prix avec accumulation des options (manque de temps pour une solution élégante).
- **Solution** : Script PHP monolithique dans la page de récapitulatif avec hardcoding partiel des tarifs.

## Phase #3

### Contraintes Humaines

- **Problème** : Réintégration de Damien dans le flux de travail après son absence. Nécessité de le mettre à niveau sur l'état du code et les choix de la Phase 2.
- **Solution** : Session de briefing dédiée. Utilisation intensive de Git et des branches pour isoler le travail de chacun et faciliter la revue de code avant fusion. Commentaires clairs dans le code.
- **Problème** : Difficulté initiale à coordonner le travail sur des fonctionnalités interdépendantes (ex : validation JS et envoi PHP, calcul de prix JS et données PHP).
- **Solution** : Adoption stricte de branches Git par fonctionnalité (`feature/dark-mode`, `feature/profile-js` etc.). Points de synchronisation réguliers (mini-réunions) pour discuter des interfaces entre les parties JS et PHP. Fusion progressive sur une branche `develop` avant de merger sur `main`.

### Défis Organisationnels

- **Problème** : Gestion de la complexité croissante du code JavaScript et risque de "code spaghetti".
- **Solution** : Organisation du code JS en fichiers distincts par fonctionnalité/page. Utilisation de fonctions bien définies. Ajout de commentaires pour expliquer la logique complexe. Convention de nommage pour les variables et fonctions.
- **Problème** : Assurer la cohérence entre les calculs/validations faits en JS (Phase 3) et ceux faits en PHP (Phase 2), notamment pour le prix.
- **Solution** : S'assurer que les données de base utilisées par JS (prix des options, règles de validation) sont les mêmes que celles utilisées par PHP (souvent en les passant via des attributs

`data-*` ou un objet JSON injecté dans la page par PHP). Le calcul final côté serveur (récapitulatif) reste la référence.

### Difficultés Techniques

- **Problème** : Débogage du code JavaScript, notamment les interactions avec le DOM et la gestion des événements.
- **Solution** : Utilisation intensive des outils de développement du navigateur (console, debugger, inspecteur DOM). Utilisation de `console.log` pour tracer l'exécution. Tests unitaires informels des fonctions JS critiques.
- **Problème** : Assurer la persistance correcte du thème via les cookies (gestion de la lecture/écriture du cookie, application du thème au chargement de chaque page).
- **Solution** : Création de fonctions utilitaires JS pour lire et écrire les cookies. Exécution d'un script au chargement de chaque page pour lire le cookie et appliquer le thème correspondant avant l'affichage complet.
- **Problème** : Gestion du merge des branches Git, surtout lorsque plusieurs membres modifient les mêmes fichiers (ex : CSS ou templates PHP pour intégrer le JS).
- **Solution** : Communication avant de commencer à travailler sur des fichiers partagés. Merges fréquents de la branche `develop` dans les branches de fonctionnalités pour réduire les conflits. Résolution collaborative des conflits de merge si nécessaire.

## Phase #4

### Contraintes Humaines

- **Problème** : Difficultés personnelles impactant la disponibilité et la concentration de certains membres. Jibril a dû gérer des soucis personnels, tandis que Jérémie, travaillant les week-ends en entrepôt et confronté à des problèmes de santé affectant ses jambes, a eu une capacité de travail réduite, notamment pour l'apprentissage approfondi de JavaScript. Damien a continué sa réintégration progressive dans le projet.
- **Solution** : Communication transparente au sein de l'équipe sur les contraintes de chacun. Réajustement des attentes et des charges de travail lorsque possible. Priorisation des tâches essentielles pour assurer la complétion des fonctionnalités clés de la Phase 4.

### Défis Organisationnels

- **Problème** : La période (vacances pour certains, engagements personnels variés) a rendu la coordination synchrone plus complexe. La pression des échéances académiques parallèles a également pesé sur l'équipe.
- **Solution** : Pour pallier ces défis, nous avons optimisé notre gestion de projet asynchrone. La création et l'utilisation rigoureuse d'un tableau d'issues sur GitHub ont permis une gestion des tâches grandement améliorée, réduisant le besoin d'appels fréquents et s'adaptant mieux à nos contraintes individuelles. Cela a permis à chacun de visualiser l'avancement et les priorités.

### Difficultés Techniques

- **Problème** : La mise en œuvre des requêtes asynchrones (AJAX) a présenté plusieurs défis :
  - Pour Jérémie, les contraintes de santé ont limité le temps disponible pour maîtriser pleinement les subtilités de JavaScript avancé, notamment la gestion des promesses et le débogage des appels asynchrones.
  - Pour l'ensemble de l'équipe, assurer la robustesse des échanges de données JSON entre le client (JavaScript) et le serveur (PHP), en gérant correctement les erreurs et les cas limites.
  - La gestion de l'état côté client après des mises à jour asynchrones (par exemple, s'assurer que l'interface reflète fidèlement les données serveur sans rechargement complet).

- Le débogage des scripts PHP appelés en AJAX, qui ne fournissent pas toujours des erreurs aussi directes qu’une page PHP classique.
- **Solution :**
  - Sessions de pair-programming et partage de connaissances au sein de l’équipe pour surmonter les blocages techniques. Damien et Jibril ont pu apporter un soutien sur les aspects JavaScript plus complexes.
  - Utilisation systématique de la console du navigateur pour tracer les requêtes et les réponses AJAX, et des outils de débogage PHP côté serveur.
  - Standardisation du format des réponses JSON du serveur (par exemple, inclure toujours un statut de succès/échec et un message).
  - Tests rigoureux des fonctionnalités asynchrones pour identifier et corriger les comportements inattendus.

## Conclusion

### Conclusion de la Phase #1

Ce rapport présente les grandes lignes du travail réalisé durant la Phase #1, les tâches réparties entre les membres, ainsi que les difficultés rencontrées et les solutions apportées. Nous continuerons à mettre à jour ce document pour refléter l’évolution du projet au fil des phases.

### Conclusion de la Phase #2

Malgré un contexte difficile (équipe réduite à 2 membres, surcharge de travail externe), les fonctionnalités de base sont opérationnelles. Les choix techniques reflètent des compromis pragmatiques :

- La personnalisation des voyages fonctionne mais manque de flexibilité (options prédéfinies)
- Le CSS a été uniformisé mais certaines pages manquent de cohérence visuelle

L’expérience a renforcé notre capacité à prioriser sous pression, bien que la dette technique accumulée nécessitera une attention particulière lors de la Phase 3.

Pour la prochaine phase, nous prévoyons de :

1. Finaliser la page de paiement
2. Implémenter les fonctionnalités de mise à jour en temps réel des prix et durées
3. Améliorer l’expérience utilisateur avec des filtres de recherche avancés
4. Renforcer la sécurité de l’application

L’architecture mise en place nous permet maintenant d’avoir une base solide pour développer la suite du projet.

### Conclusion de la Phase #3

La Phase 3 a permis d’enrichir considérablement l’expérience utilisateur en ajoutant une couche d’interactivité JavaScript significative. Les actions courantes comme la validation de formulaires, la modification de profil, le tri de résultats ou l’estimation de prix se font désormais sans rechargement de page, rendant l’application plus fluide et moderne. Le retour de Damien et l’adoption d’une gestion de branches Git plus structurée ont amélioré la collaboration malgré la complexité accrue. Les défis techniques liés à JavaScript et à la cohérence client/serveur ont été relevés. Le site est maintenant prêt pour l’étape finale : l’introduction des requêtes asynchrones pour une communication serveur sans rechargement complet.

### Conclusion de la Phase #4

Cette dernière phase a été intense, marquée par l’implémentation des fonctionnalités asynchrones qui modernisent l’interaction utilisateur et par la finalisation globale du projet. Malgré les contraintes personnelles, la période d’examens et les défis techniques inhérents aux technologies employées, l’équipe

a su faire preuve de résilience et d'organisation, notamment grâce à une meilleure gestion des tâches via GitHub. Les objectifs principaux de la phase 4 ont été atteints, offrant une application plus réactive et complète. Nous avons fait de notre mieux pour livrer un produit fonctionnel et conforme aux attentes, compte tenu du temps et des ressources disponibles.

## Bilan du projet

Au terme de ces quatre phases, le projet a représenté une expérience d'apprentissage significative en matière de développement web full-stack, de gestion de projet en équipe et d'adaptation face aux imprévus. Chaque phase a apporté son lot de défis et de réussites, contribuant à renforcer nos compétences techniques et organisationnelles. Le site web développé, bien que perfectible, constitue une réalisation concrète de nos efforts collectifs. (Si on avait été payés, on serais super riches.)