

# Rapport de Projet

## Informatique 4

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

Dirigé par : LE BRETON Caryl

3 mai 2025

## Table des matières

<b>Introduction</b>	<b>3</b>
<b>Planning</b>	<b>3</b>
Phase #1 . . . . .	3
<b>Répartition des Tâches</b>	<b>3</b>
Phase #1 . . . . .	3
<b>Problèmes Rencontrés et Solutions Apportées</b>	<b>3</b>
Phase #1 . . . . .	3
Problèmes Humains . . . . .	3
Problèmes Organisationnels . . . . .	3
Problèmes Techniques . . . . .	3
<b>Conclusion</b>	<b>3</b>
<b>Planning</b>	<b>4</b>
Phase #2 . . . . .	4
<b>Répartition des Tâches</b>	<b>4</b>
Phase #2 . . . . .	4
<b>Format des Données</b>	<b>4</b>
Organisation des Fichiers de Données . . . . .	4
<b>Fonctionnalités Implémentées</b>	<b>5</b>
Système d'Authentification . . . . .	5
Affichage des Voyages . . . . .	5
Vue Détaillée et Personnalisation des Voyages . . . . .	5
Récapitulatif et Paiement . . . . .	5
<b>Problèmes Rencontrés et Solutions Apportées</b>	<b>5</b>
Phase #2 . . . . .	5
Contraintes Humaines . . . . .	5
Défis Organisationnels . . . . .	5
Difficultés Techniques . . . . .	6
<b>Conclusion de la Phase #2</b>	<b>6</b>
<b>Phase #3 : Dynamisme Côté Client avec JavaScript</b>	<b>6</b>
Planning Réalisé (Phase #3) . . . . .	6
Répartition des Tâches (Phase #3) . . . . .	6
Fonctionnalités Implémentées (Phase #3) . . . . .	7
Problèmes Rencontrés et Solutions Apportées (Phase #3) . . . . .	8
Contraintes Humaines . . . . .	8
Défis Organisationnels . . . . .	8
Difficultés Techniques . . . . .	8
Conclusion (Phase #3) . . . . .	9

## 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.

## Répartition des Tâches

### Phase #1

La répartition des tâches entre les membres du groupe est la suivante :

- **Jérémie** : Conception de l'interface utilisateur et création des maquettes.
- **Jibril** : Développement du front-end et intégration en HTML/CSS.
- **Damien** : Coordination du projet, gestion de version (Git) et rédaction du rapport.

## 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.

## Conclusion

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.

## Planning

### 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.

## Répartition des Tâches

### Phase #2

Répartition adaptée en raison de l'absence temporaire de Damien et des contraintes personnelles :

- **Jérémie** :
  - Création/modification des templates (détail, récapitulatif, liste voyages)
  - Conception des fichiers JSON et logique de personnalisation
  - Ébauche de la page de paiement (non finalisée)
  - Gestion des données dynamiques (prix, durée)
  - Système de recherche basique
- **Jibril** :
  - Transformation HTML vers PHP (inscription, connexion, profil)
  - Implémentation des sessions utilisateurs
  - Refonte CSS globale et responsive design
  - Tests de compatibilité et corrections d'urgence

## Format des Données

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.

## Fonctionnalités Implémentées

### 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.

## Problèmes Rencontrés et Solutions Apportées

### 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.

## 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.

## Phase #3 : Dynamisme Côté Client avec JavaScript

### Planning Réalisé (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.

### Répartition des Tâches (Phase #3)

Avec le retour de Damien, la répartition a été revue :

- **Jérémie** :
  - Mise à jour du rapport PDF.
  - Implémentation du calcul dynamique du prix en JavaScript sur la page de détail du voyage.
  - Développement de la fonctionnalité "Panier" (logique de stockage des voyages en cours de modification/consultation, probablement via session ou localStorage).
  - Implémentation du tri côté client sur la page de recherche de voyages.
- **Jibril** :
  - Amélioration de la page de profil utilisateur : ajout de l'édition champ par champ en JavaScript (activation/désactivation, validation/annulation sans rechargement).
  - Amélioration de la page administrateur : simulation de la mise à jour des propriétés utilisateur avec délai visuel en JavaScript.

- **Damien :**
  - Implémentation de la validation des formulaires (inscription, connexion) côté client en JavaScript (vérification des champs, messages d'erreur sans rechargement, compteur de caractères, visibilité mot de passe).
  - Mise en place du changement de charte graphique (ex : mode sombre/clair) avec persistance via cookie.
  - Gestion des merges des branches Git et coordination technique.

## Fonctionnalités Implémentées (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.

## Problèmes Rencontrés et Solutions Apportées (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.

## Conclusion (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.