

Rapport de Projet – Phase 1

1. Répartition des tâches

Membre du groupe	Tâches principales
Church William	Pages circuits et inscription, organisation des fichiers
Kuganesan Arun	Début du site, choix des couleurs, pages d'accueil et admin
Bouhou Haytham	Pages de profil et de connexion

2. Problèmes rencontrés

2.1 Problèmes humains

- Désaccord sur les couleurs du site (Arun voulait les changer, les autres non)
- Décalage dans les moments de travail, compliquant la communication

2.2 Problèmes organisationnels

- Travail collaboratif difficile via Git : chacun travaillait localement en parallèle
- Risques de conflits lors des modifications simultanées

2.3 Problèmes techniques

- Conflits de version et de synchronisation sur Git
 - Fusion difficile en cas de travail simultané
-

3. Solutions et améliorations

- Organisation du travail : chacun évite de modifier les mêmes fichiers en même temps
 - Choix collectif d'une palette de couleurs pour limiter les désaccords
 - Meilleure planification des sessions de développement
-

4. Archivage

- Le projet est versionné via Git
 - Les mises à jour sont enregistrées par commit, avec création de branches si besoin
 - Certaines versions sont aussi conservées en formats ODT / DOCX pour le rapport
-

5. Mise à jour

- Date de mise à jour : 16/02/2025
-

Rapport de Projet – Phase 2

1. Répartition des tâches

Membre du groupe	Tâches principales
Church William	PHP (toutes les pages), CSS, gestion JSON
Kuganesan Arun	HTML, CSS, gestion des fichiers JSON de voyages, PHP (connexion)
Bouhou Haytham	JSON (utilisateurs et voyages), gestion des images

2. Format des données

2.1 `utilisateurs.json`

- `id`, `nom`, `prénom`, `email`, `mot_de_passe`, `date_naissance`, `admin`, `historique`

2.2 `voyage01.json`, `voyage02.json`, etc.

- `titre`, `dates`, `étapes`, `prix_total`, `statut`

2.3 `paiements.json`

- Coordonnées bancaires (fictives), utilisateur, voyage, date_transaction

2.4 Pourquoi JSON ?

- Simple, léger, natif en PHP
 - Parfait pour des structures arborescentes comme les voyages ou l'historique utilisateur
-

3. Fonctionnalités Implémentées

- Inscription et connexion sécurisées (hash + vérif)
 - Gestion des rôles (admin / utilisateur)
 - Liste synthétique des voyages
 - Barre de recherche par titre
 - Personnalisation des voyages (étapes et options)
 - Saisie simulée du paiement et enregistrement
-

4. Tests et validation

- Tests utilisateurs créés pour chaque rôle
 - Voyages tests simulés en JSON
-

Rapport de Projet – Phase 3

1. Répartition des tâches

Membre du groupe	Tâches réalisées
Church William	Tri/filtre dynamique JS, profil JS, récapitulatif, cookie de thème, édition profil
Kuganesan Arun	JS formulaires, CSS graphique, gestion thème dynamique
Bouhou Haytham	Panier JS+PHP, suppression AJAX, page admin + latence simulée

2. Fonctionnalités dynamiques (JS)

2.1 Thème clair/sombre

- Basculable sans rechargement
- Sauvegarde dans un cookie

2.2 Formulaires (inscription / connexion)

- Vérification JS en direct
- Messages d'erreur dynamiques
- Icône œil + compteurs mot de passe

2.3 Édition directe du profil

- Champs désactivés par défaut
- Boutons d'activation individuels
- Soumission uniquement si un champ est modifié

2.4 Admin : attente simulée

- Bouton admin désactivé temporairement
- Affichage d'un délai de traitement

2.5 Recherche/tris des voyages

- Tri sans rechargement : prix, durée, etc.

2.6 Prix dynamique (JS)

- Recalcul côté client à chaque option modifiée

3. Panier (PHP + JS)

- Ajout automatique lors de la validation
- Affichage dans le header (menu déroulant)
- Suppression via AJAX, compteur dynamique

4. Organisation du code

- JavaScript dans fichiers externes
- CSS thématisé (à unifier en phase 4)
- Sécurisation de `panier.php`, `supprimer_panier.php`

5. Tests

- Formulaires, panier, prix dynamiques
- Affichage, accessibilité, cookies
- Comptes de test créés

6. Bilan

- Projet plus ergonomique, réactif et structuré
- Architecture modulaire et claire

Rapport de Projet – Phase 4

1. Répartition des tâches

Membre du groupe	Tâches réalisées
Church William	Ajout et fonctionnement du bouton Bannir, mutualisation du code, sauvegarde des données
Kuganesan Arun	Correction formulaire connexion, CSS du profil admin, sécurisation des fichiers JSON
Bouhou Haytham	Chargement dynamique des options de voyage, correctifs JS

2. Fonctionnalités ajoutées

2.1 Mise à jour du profil utilisateur

Cette fonctionnalité avait déjà été mise en place en **phase 3**. Elle a été conservée et validée dans cette dernière phase.

Lorsqu'un utilisateur modifie un champ (nom, prénom, mot de passe...), la mise à jour est transmise via une requête AJAX.

- ☒ En cas de succès : les nouvelles données sont affichées dans le formulaire.
- ☒ En cas d'échec : les anciennes données sont restaurées automatiquement.

2.2 Modification depuis la vue administrateur

Également réalisée lors de la **phase 3**, cette fonctionnalité a été consolidée :

- Ajout du bouton « **Bannir** » (champ `banni` dans le JSON utilisateur).

- Affichage d'une **animation de chargement** pendant le traitement (simulateur de latence activé).
-

2.3 Chargement dynamique des options dans un voyage

- Le client JS envoie une requête AJAX vers `voyage.php?options=...`
- Le serveur lit le JSON (`voyage01.json`, etc.) et renvoie les listes d'options.
- JavaScript reconstruit dynamiquement les `<select>` pour chaque catégorie (hébergement, activité...).






Ce fonctionnement rend l'ajout d'options plus flexible sans toucher au HTML.

2.4 Calcul asynchrone du prix total du voyage






- Lors d'un changement d'option ou du nombre de personnes, une requête AJAX POST est envoyée.
- Le prix est recalculé côté serveur et renvoyé instantanément.
- Il est mis à jour dynamiquement sans rechargement de la page.

Cela garantit que l'utilisateur voit toujours le bon tarif.

2.5 Finalisation et correctifs

-  **Mutualisation du code** (ex : chargement JSON, gestion erreurs).
 -  **Protection des fichiers JSON** contre l'accès direct.
 -  **Conservation de l'e-mail saisi** dans le formulaire de connexion si le mot de passe est erroné.
 -  **Suppression automatique d'un voyage en cours de modification** pour éviter les doublons dans le panier.
 -  **Corrections de bugs** : restauration des champs, synchronisation des données du panier, cohérence affichage/sessions.
-

3. Tests et validation

-  Tests AJAX profil utilisateur (succès + échec)
 -  Tests admin : bannir un utilisateur, animation de chargement
 -  Tests sur restauration automatique (options, date, personnes)
 -  Tests sur calcul dynamique du prix
 -  Sécurité JSON, intégrité session
-

4. Bilan de la phase 4

Cette dernière phase a permis de :

- finaliser toutes les fonctionnalités prévues,
- stabiliser le code existant,
- améliorer l'expérience utilisateur,
- renforcer la sécurité du projet.

Le rapport a été mis à jour et **intégré dans le même document PDF** que les phases précédentes, sans créer de fichier séparé, conformément aux consignes.