

Répartition des tâches :

Louis	Ethan	Jhoneyd
Formulaire d'inscription formulaire de connexion page de profil A propos	Page d'accueil ensemble des voyages	Page administrateurs Rapport

Phase 1:

Planning :

31 Janvier : Mise en places de la charte graphique

7 Février : Mise en accord de l'aspect du site et des pages de celui-ci

7-16 Février : création des pages

1/Description du travail

Au cours de cette première phase, nous avons concentré nos efforts sur la partie visible du site, c'est-à-dire la partie client. L'objectif de cette phase était de mettre en place les fondations du projet en définissant son univers graphique et en créant les premières pages statiques en HTML et CSS. Pour ce faire, nous avons commencé par choisir un nom et un thème à notre agence de voyages afin de lui donner sa propre identité.

Dans un deuxième temps, nous avons établi une charte graphique précisant l'harmonisation des couleurs choisies et également leurs usages (titres, textes, boutons, fonds, etc...). Dans un troisième temps, nous avons réalisé plusieurs pages statiques : une page d'accueil présentant le site, une page de recherche d'un voyage avec des filtres, une page

d'inscription, une page de connexion, une page de profil utilisateur, ainsi qu'une page d'administration. Chacune d'elle a été développée séparément en HTML et allie une feuille de style CSS commune pour assurer une cohérence visuelle. Cette étape nous permet de construire notre site afin d'établir les solides bases aux fonctionnalités dynamiques qui viendront dans les phases ultérieures.

2/Problème rencontré

Dans l'ensemble le groupe n'a pas rencontré de problème au début l'organisation était difficile mais après quelques réunions le groupe a réussi à se mettre en route et à se répartir les tâches afin d'avancer. Le groupe était en désaccord sur ce à quoi devait ressembler le site mais après plusieurs réunions nous sommes parvenus à un style épuré avec une interface utilisateurs minimaliste afin de faciliter son utilisation. Techniquement tous les membres du groupe n'étaient pas familiers avec les langages utilisés mais ont pu demander de l'aide aux autres membres afin de réaliser les tâches et de mieux comprendre les langages utilisés.

.

Phase 2:

1/Description du travail

Durant cette phase, l'intégralité de la partie serveur a constitué notre cœur de cible, en visant à dynamiser le site et le rendre interactif en gestion de données back-end. Dans un premier temps, il nous a paru important de mettre en œuvre un système d'inscription et de connexion des utilisateurs pour vérifier les informations renseignées lors de l'inscription puis les mettre à proximité d'un fichier JSON destinés à les stocker. Lors de l'authentification de l'utilisateur le site vérifie ses identifiants, ouvre alors une session PHP pour que l'utilisateur bénéficie d'un espace personnel.

De là, nous nous sommes penchés sur la mise en page des voyages, le site a la possibilité d'afficher une liste de voyages, où s'affichent les informations essentielles : le titre, les dates, le prix, le nombre d'étapes. L'utilisateur peut cliquer directement sur un voyage, qui renvoie alors vers une page détaillée où il peut construire son séjour, libre à lui de choisir pour chaque étape : hébergement, restauration, activité, transport...

La partie paiement a également été intégrée. Bien qu'elle soit simulée, elle est conçue dans un véritable processus : l'utilisateur remplit un formulaire avec ses coordonnées bancaires, ces données sont envoyées à un script de vérification, et selon la réponse (succès ou échec), on renvoie vers la page de confirmation ou d'erreur. En cas de succès, le voyage est configuré avec les données de paiement.

Enfin, notre interface administrateur permet d'accéder à la liste des utilisateurs inscrits et offre des filtres et la possibilité d'agir sur leur statut (passer un statut à VIP par exemple ou bloquer le compte). Cette interface est réservée aux comptes avec le rôle « admin ».

2/ Problèmes rencontrés

Le principal souci a été de comprendre le fonctionnement du système de paiement simulé. Le script fourni vérifie les coordonnées bancaires et redirige automatiquement selon que le paiement est accepté ou refusé. Il a donc fallu bien organiser les différentes pages PHP : formulaire de saisie, traitement, redirection en cas d'erreur ou de succès, puis enregistrement de la commande. Au début, c'était flou, surtout sur le lien entre les scripts et la logique de redirection. Après plusieurs tests, on a réussi à bien suivre le chemin complet du paiement, et à tout rendre fonctionnel.

Le second problème rencontrés au cours de cette phase a été la mise en place et la compréhension de la gestion des sessions en PHP. Au départ, il n'était pas évident de savoir comment maintenir l'état de connexion d'un utilisateur entre les différentes pages du site. La manipulation des variables de session, ainsi que la mise en place des redirections en fonction du statut de l'utilisateur (connecté ou non), nous a demandé plusieurs essais et ajustements.

Phase 3 :

1/Description du travail

À cette troisième phase, l'objectif principal visé a été de rendre le site du côté client plus interactif, tirant parti des possibilités d'interaction disponibles en JavaScript. De cette façon, l'expérience utilisateur a été améliorée sans nécessiter le rechargement des pages.

Au départ, nous avons mis en place une fonction de changement de thème visuel (mode clair/sombre) qui repose sur le chargement dynamique de fichiers CSS et qui sauvegarde le choix dans un cookie pour qu'il soit automatiquement appliqué lors des connexions ultérieures.

Puis nous avons enrichi tous les formulaires du site d'une validation côté client (s'inscrire, se connecter, payer). En effet chaque champ est testé avant l'envoi sur le serveur pour éviter les éventuelles erreurs côté serveur. De plus nous avons prévu d'indiquer des éléments comme par exemple la possibilité de rendre le mot de passe bien visible ou bien le nombre de caractères qu'il reste encore avant que l'on ne soit saturé.

Enfin sur la page de profil les champs ne sont plus modifiables par défaut. L'utilisateur peut activer à sa convenance l'édition champ par champ, valider ou bien abandonner le changement sans aucune requête HTTP tant que le formulaire n'est pas entièrement soumis.

Vis-à-vis de l'administrateur, nous avons simulé le temps de traitement d'un changement de statut (VIP, blocage). Au moment où une action est effectuée, l'élément se désactive quelques secondes avant de se mettre à jour, ce qui va dans le sens d'un comportement de traitement serveur.

Enfin sur les pages voyages, nous avons ajouté la possibilité de faire un tri dynamique au coût (au prix, à la date...) après une recherche, et de recalculer en temps réel le coût d'un voyage au moment où un utilisateur modifie une option (hébergement, activités, transport...).

2/Problèmes rencontrés

Le premier obstacle a résidé dans la gestion dynamique des formulaires, à savoir l'enchaînement des événements JS (édition , validation , annulation), qui nécessitait de penser la logique afin d'éviter les conflits.

L'intégration du changement de thème a aussi posé problème : charger dynamiquement une feuille de style sans perturber l'affichage a nécessité plusieurs ajustements.

La mise en place du recalcul de prix s'est avéré bien plus complexe que prévu, car il fallait gérer plusieurs variables (nombre de personnes, options par étape), tout en assurant un affichage cohérent.

Enfin, la modularisation des scripts JavaScript pour garder un code propre et indépendant selon les pages (profil, admin, voyages) a demandé une bonne architecture et l'usage de fonctions encapsulées.

Phases 4 :

1/ Description du travail

Pour cette quatrième étape, nous avons pour l'essentiel axé nos travaux sur la mise en œuvre de requêtes asynchrones, pour les enjeux de fluidité et d'expérience utilisateur. L'objectif essentiel du projet était d'envisager la possibilité de travailler les contenus des pages sans avoir à les recharger pour proposer ainsi une navigation à la fois moderne et plus agréable.

À ce stade, il est approprié de dire que nous avons utilisé des requêtes de type AJAX pour répondre à notre problème : si, par exemple, l'utilisateur modifie ses informations dans la page de son profil, les données sont désormais envoyées en arrière-plan vers le serveur, et une fois la validation du changement faite, le champ est mis à jour sans recharge de l'intégralité de la page, renforçant ainsi la fluidité de l'expérience sur le site.

Sur le volet administrateur aussi, cette méthode a été mise en place afin de permettre à un administrateur de faire évoluer un utilisateur (le faire devenir VIP ou le bloquer) d'une manière dynamique, sans recharger la page du navigateur. Ainsi, une fois l'action effectuée, le statut est mis à jour, ce qui donne l'impression que le changement est fait en temps réel.

Pour la gestion du panier, tout a été fait pour que, pendant que l'utilisateur personnalise un voyage (activités, hébergement, transport, ...), toutes ses modifications soient prises en compte dans le panier lorsqu'un changement s'opère, sans recharger la page. Tout est mis en place pour revenir plus tard tout en ayant gardé, dans le panier, ses préférences. Le panier évolue après chaque modification, sans qu'il soit nécessaire de confirmer l'achat.

Enfin, à l'aide d'un bouton de mise à jour asynchrone, il est possible de rafraîchir toute autre donnée, comme la liste des utilisateurs ou des voyages, pour avoir accès à ce qui est le plus récent sur le serveur, sans qu'il soit besoin de quitter la page de travail.

2/ Problèmes rencontrés

Au cours de cette phase, ce qui s'est avéré pertinent n'a pas été d'ajouter, mais d'améliorer les fonctionnalités et de les rendre interactives et ergonomiques au moyen de l'asynchrone.

Le véritable problème de la bonne construction des requêtes asynchrones est mis au jour avec l'utilisation d'AJAX. Nous avons dû corriger nombre de scripts PHP pour qu'ils retournent des réponses claires au format JSON et non sous n'importe quelle forme. Il a fallu également gérer efficacement les messages d'erreur au cas où l'utilisateur pourrait être bloqué dans sa navigation.

Un autre problème, celui de la gestion du JavaScript, se posait. Chaque page ayant ses propres fonctions ne doit pas interférer avec les autres, ce qui exige du temps pour bien gérer le code afin que tout fonctionne sur le site.

Enfin, la gestion des sessions et des permissions pour les requêtes AJAX a impliqué quelques aménagements pour que les utilisateurs non connectés n'aient pas la possibilité d'effectuer des requêtes éventuellement non autorisées.