

Document planning

1) Répartition des tâches

Pour la répartition des tâches, nous avons essayé de rendre ça égalitaire, en se donnant chacun des pages à faire chacun de notre côté, puis de revoir notre travail en classe ...

-C'est-à-dire que la page 1 d'accueil par défaut avec la présentation du site et la la page 2 pour la présentation du site qui reprend notre nom et notre thème + un champ de recherche rapide intégré, et la page 7 type 'administrateur' avec une liste d'utilisateurs inscrits et des boutons permettant de modifier une propriété de chaque utilisateur (ex : client VIP qui aurait une réduction des prix, bannissement du client qui ne pourrait plus acheter de voyages, ...) ont été réalisés par Ninjbaatar.

-La page 3, la page spécifique pour rechercher des voyages avec plusieurs champs de filtrage (dates, lieux, options, ...) et la page 4 celle avec un formulaire pour s'inscrire, et le document de conception ont été réalisés par Lorenzo.

-La page 5 avec un formulaire pour se connecter lorsque l'on est déjà inscrit sur le site et la page 6 pour afficher le propre profil d'un utilisateur connecté avec des boutons pour modifier les différents champs (bouton 'modifier' ou avec un icône 'crayon', ...), et le document de planning ont été réalisés par Matéo.

On peut donc dire une répartition égalitaire : 34/33/33 pour Ninj/Lorenzo/Matéo.

2) Problèmes rencontrés

- On avait un problème d'espace blanc sur l'affichage de notre site en haut et en bas.
- Mais aussi on avait des problèmes d'espaces et les caractères n'étaient pas à leurs places.

3) Solutions apportées

-Nous avons résolu le problème d'espace blanc en ajoutant des styles directement dans le HTML, sans toucher au fichier CSS.

Concrètement, nous avons :

- a) Retiré les marges et le padding par défaut en définissant « margin:0; padding:0; » sur <html> et <body> via des styles inline.
- b) Utilisé « min-height: 100vh » sur < body> pour s'assurer que le contenu occupe toute la hauteur de l'écran.

- c) Employé un conteneur « Flexbox » (via des styles inline) pour centrer verticalement et horizontalement le contenu, ce qui réduit l'espace blanc autour des éléments.

-Et pour résoudre le problème de caractères et d'espaces nous avons obtenu les pages d'inscription et de connexion sans toucher au fichier CSS en appliquant directement des styles via des attributs inline dans le HTML.

Concrètement, nous avons :

- a) Utilisé des conteneurs « flex inline » pour centrer le contenu (via « display: flex; justify-content: center; align-items: center; min-height: 100vh; ») afin de réduire l'espace blanc autour du formulaire.
- b) Ajouté des styles inline aux éléments (comme aux inputs et boutons) pour ajuster le padding, les marges, les bordures, et la taille de police, ce qui améliore la visibilité et l'apparence des champs.
- c) Conservé le fichier CSS inchangé en apportant toutes les modifications de présentation directement dans le HTML grâce aux styles inline.

Phase 2

1) Répartition des tâches :

Pour cette répartition des tâches en phase 2 , Lorenzo a fait les programmes à propos de l'inscription d'un utilisateur, sa connexion ainsi que le programme administrateur.

Ninjbaatar s'est occupé des programmes d'affichages des voyages, ainsi que l'affichage en liste, aussi le voyage détaillé et le programme qui récapitule le voyage du client.

Matéo a fait le programme qui s'occupe du paiement du client, ainsi que l'interface utilisateur du profil du client, et les documents de conception et de planning.

On peut donc dire une répartition égalitaire : 33/34/33 pour Ninj/Lorenzo/Matéo.

2) Problèmes rencontrés :

-En premier, il y a eu un problème à propos de la mauvaise récupération de l'ID de l'utilisateur en session, c'est-à-dire lors de l'affichage des voyages payés sur la page de profil (profil.php), l'ID utilisateur stocké en session n'était pas toujours bien récupéré, ce qui empêchait d'afficher les voyages de l'utilisateur connecté.

-En second, une Mauvaise validation des coordonnées bancaires, c'est-à-dire lors de la soumission du formulaire de paiement (paiement.php), certains utilisateurs saisissaient un numéro de carte incorrect (trop court ou avec des espaces), ce qui entraînait des erreurs lors de l'envoi au script de validation.

- Et pour finir, un affichage incorrect des détails du voyage, lorsqu'un utilisateur cliquait sur un voyage dans la liste de la page de profil, il était parfois redirigé vers une page vide au lieu des détails du voyage.

3) Solutions apportées :

-La cause du premier problème était que la session n'était pas toujours bien démarrée au début du script (session_start(); manquant ou mal placé). Donc la solution a été d'ajouter un session_start(); en tout début du fichier avant toute sortie HTML. Cela garantit que la session est bien démarrée et que l'utilisateur est authentifié avant d'afficher ses voyages.

```
<?php
```

```
session_start();
```

```

if (!isset($_SESSION['user_id'])) {

    header("Location: login.php");

    exit();

}

?>

```

- La cause du second problème était qu'il y avait aucune vérification côté serveur pour s'assurer que le numéro de carte était bien composé de 16 chiffres uniquement. Donc comme solution on a ajouté une validation en PHP pour supprimer les espaces et vérifier la longueur du numéro. Cela empêche les erreurs dues aux espaces ou aux numéros trop courts/longs.

```

$numero_carte = str_replace(' ', '', $_POST['numero_carte']);

if (!preg_match('/^\d{16}$/', $numero_carte)) {

    echo "Numéro de carte invalide.";

    exit();

}

```

- Et la cause du dernier problème était que l'ID du voyage passé en paramètre (?id=...) n'était pas bien récupéré ou était absent. La solution a été un ajout d'une vérification pour s'assurer que l'ID du voyage est bien présent dans l'URL avant d'exécuter la requête SQL.

```

if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {

    echo "Voyage introuvable.";

    exit();

}

$id_voyage = intval($_GET['id']);

```

Phase 3 :

1) Répartition des tâches :

Pour cette répartition des tâches de la phase 2, Lorenzo s'est occupé des deux premiers programmes dont la fonctionnalité « clair/sombre » ..., et du formulaire, pour les 3 autres programmes c'est Neji qui a donc fait le « profil complet », la fonctionnalité « administrateur » et celle de la recherche de voyage, et pour finir les deux derniers programmes ont été faits par Matéo dont le prix dynamique et le panier et les documents de conception et de planing.

On peut donc dire une répartition égalitaire : 33/33/34 pour Ninj/Lorenzo/Matéo.

2) Problèmes rencontrés :

En premier, le site qui ne supportait pas le fait qu'on navigue plusieurs entre les fichiers



-En second, le manque de cohérence entre le prix affiché en JavaScript et celui calculé en PHP.

Le problème était que le prix affiché dynamiquement en JavaScript était parfois différent du prix final calculé côté serveur, car les tarifs étaient définis séparément (une fois dans prix_dynamique.js et une autre fois dans recapitulatif.php).

-Ensuite, la mise à jour du prix dynamique sans requête HTTP.

Le problème était que la mise à jour du prix devait se faire instantanément sans requête HTTP ni rechargement de la page.

-Puis, l'ajout automatique du voyage au panier.

Le problème était qu'il fallait ajouter un voyage au panier automatiquement dès que l'utilisateur modifie une option ou affiche le récapitulatif avant paiement.

-Et, la gestion du stockage côté serveur.

Le problème était que le panier devait être exclusivement géré en PHP, car les utilisateurs doivent être connectés pour y accéder.

3) Solutions apportées :

-Solution apportée : Aucune à part la patience.

-Solution apportée : Nous avons centralisé les tarifs dans un fichier unique `data/tarifs.json`, qui est maintenant utilisé à la fois par le JavaScript (`prix_dynamique.js`) et par PHP (`recapitulatif.php`).

--En PHP : Les tarifs sont chargés via `json_decode(file_get_contents("data/tarifs.json"), true)`.

--En JavaScript : Les tarifs sont récupérés avec `fetch("data/tarifs.json").then(response => response.json())`.

Et donc le résultat : Le prix affiché dans `voyage_detaille.php` et celui affiché dans `recapitulatif.php` sont toujours parfaitement synchronisés !

-Solution apportée : Nous avons créé un fichier JavaScript séparé (`js/prix_dynamique.js`) qui :

--Écoute les modifications des inputs (`onchange`, `oninput`).

--Recalcule le prix immédiatement sans requête serveur.

--Met à jour l'affichage avec `document.getElementById("prix_total").innerText = prixTotal + " €";`.

Et donc le résultat : Lorsque l'utilisateur modifie une option, le prix change instantanément sans aucun rafraîchissement !

-Solution apportée : Nous avons créé un formulaire caché dans `voyage_detaille.php` qui est soumis automatiquement grâce à `js/panier.js`.

--En PHP (`ajouter_panier.php`) : Stocke les choix du voyage en session (`$_SESSION["panier"]`).

--En JavaScript (`js/panier.js`) : Soumet le formulaire caché dès qu'une option est modifiée (`document.getElementById("form-panier").submit();`).

Et donc le résultat : Le panier est rempli automatiquement sans intervention de l'utilisateur.

-Solution apportée : Nous avons stocké le panier en session PHP (`$_SESSION["panier"]`) et empêché l'accès à `ajouter_panier.php` si l'utilisateur n'est pas connecté (`if (!isset($_SESSION["user_id"])) { die("Vous devez être connecté pour ajouter un voyage au panier."); }`).

ET donc le résultat : Seuls les utilisateurs connectés peuvent ajouter des voyages à leur panier et retrouver leurs choix plus tard !