

## **Rapport Click-Journey:**

### Phase 1:

#### Introduction:

Le projet Click-Journey consiste à créer un site web d'une agence de voyages spécialisée qui propose des séjours déjà configurés au niveau des étapes. L'objectif principal est de proposer aux utilisateurs une expérience fluide et intuitive pour explorer et réserver des séjours préconfigurés, tout en offrant la possibilité de personnaliser certaines options (hébergement, activités, transport, etc).

Pour se faire nous devons donc :

- Offrir une interface ergonomique et immersive.
- Développer un système de gestion des utilisateurs.

#### Planning :

Nous avons travaillé sur notre projet en utilisant une plateforme en ligne appelée Discord. Cet outil nous a permis de collaborer efficacement en échangeant des idées, en partageant notre code en temps réel et en réfléchissant ensemble pour résoudre les différents problèmes rencontrés. Grâce à cette organisation, nous avons pu nous corriger mutuellement et identifier les nombreuses erreurs survenues au cours de la préparation de cette première phase du projet.

Avant de commencer, nous avons d'abord organiser notre travail en plusieurs « blocs », nous avons séparé les différents fichiers HTML et nous avons un seul fichier CSS pour l'ensemble des pages. Nous avons aussi créé un dossier « media » dans lequel on retrouve les différentes images utilisées (que ce soit backgrounds ou icones). Afin d'assurer une progression efficace et organisée du projet, chaque membre de l'équipe a pris en charge des parties spécifiques du développement du site web.

Ahmed s'est occupé de la page index, celle-ci est le premier point de contact avec les utilisateurs. Dessus, nous pouvons retrouver les différentes activités proposées par notre agence « Beyond Survival » (Immersion en pleine nature (survie), stages de survie ainsi que des

escapes Game de survie). Un élément important présent sur cette page est l'animation dynamique, qui présente les trois types d'expériences disponibles. Cette animation contribue à l'aspect interactif du site et améliore l'expérience utilisateur (plus ergonomique). Encore sur cette page index on retrouve notre logo (celui-ci est cliquable et permet de revenir à l'accueil/page index) ainsi que notre agence « Beyond Survival » et notre slogan « *Survive if you can !* », de plus, il y a la possibilité de se connecter, se déconnecter, consulter son profil ainsi qu'un accès admin (tout n'est pas encore fonctionnels, seulement la partie HTML et CSS, les boutons sont dynamiques). Enfin, Ahmed a amélioré le fichier CSS principal pour harmoniser le design et assurer une cohérence visuelle sur l'ensemble du site (exemple dans le footer/bas de page, il y a le copyright).

Abdelwaheb s'est occupé de concevoir la page de présentation qui offre une vue d'ensemble sur l'identité et les valeurs de l'agence. Elle comprend deux éléments principaux :

- Une « boîte » de recherche rapide, permettant aux visiteurs d'effectuer une recherche immédiate pour trouver une expérience adaptée à leur besoins/envie. On notera aussi que celle-ci est dynamique, sa position change en fonction du mouvement de la souris de l'utilisateur (Ahmed et Abdelwaheb se sont occupés de ce changement de position).

- Une section « A propos de nous », dans lequel sont détaillés/listé l'identité de l'agence, nos missions, nos destinations ainsi que nos valeurs. Cette section aide à renforcer la crédibilité du site en donnant aux utilisateurs une meilleure compréhension des services proposés et donc montrer pourquoi nous choisir nous « Beyond Survival ».

Enfin, les boutons « Accueil » et « Rechercher » dans la nav bar sont aussi présents. On y retrouve logiquement aussi notre logo, un accès à son profil, un accès admin et la possibilité de se connecter et de se déconnecter.

Rémi, quant à lui, a pris en charge la mise en place du système de gestion des utilisateurs. Il a développé les premières bases des pages d'inscription, de connexion et de consultation de profil. Bien que ces pages ne soient pas encore fonctionnelles, elles intègrent déjà la structure HTML et CSS, ce qui permet d'avoir une première version visuelle du système. Ce travail servira donc de fondation pour l'implémentation des fonctionnalités backend (pour la prochaine phase). Dans la page profil on a donc accès aux

différentes informations personnelles tels que l'adresse mail, le nom, prénom, sexe, etc. A côté de chacune de ces informations, il y a un bouton « modifier ». Dans la page admin, on retrouve une « liste d'utilisateurs » avec les différentes infos personnels propre à chacun et des boutons tels que « modifier », « réductions », « reset mdp », « bannir » et « supprimer ». Chaque bouton est illustré par un emoji dynamique.

## Phase 2 :

Pour cette deuxième phase du projet, à la manière de la première phase, nous nous sommes séparé les tâches afin que chaque membre de l'équipe prenne en charge des parties spécifiques du développement du site web côté serveur. Avant de commencer, nous avons choisi de définir une base de données « clickjourney » gérer avec MySQL.

Ahmed s'est occupé de la création de cette base de données en utilisant MySQL/MariaDB en définissant les tables nécessaires pour stocker les informations sur les voyages, les utilisateurs et leurs rôles. Il s'est aussi occupé de la mise en place du système d'authentification :

- Inscription des utilisateurs avec une vérification des informations, connexion sécurisée avec gestion des sessions ainsi que la déconnexion et gestion des erreurs d'identification.

De plus, il y'a la gestion des rôles utilisateur (Admin/Utilisateur) : Un utilisateur simple peut consulter et réserver des voyages tandis que l'administrateur peut gérer la plateforme (ajout/modification des voyages, gestion des utilisateurs, etc). Ajout d'une pagination dans le cas où il y'aurait plus de clients.

- Gestion des options et des étapes de voyages dans « trip.php » : l'utilisateur peut modifier les étapes d'un voyage (exemple : nombre de personnes, type d'activité, etc).

- Création d'une page dans laquelle on a le récapitulatif du voyage, l'utilisateur y retrouve toutes les informations de son voyage personnalisé avant paiement.
- Filtrage des recherches : possibilités de trier ses recherches (exemple : par notes).

Abdelwaheb a travaillé sur l'affichage et l'interaction avec les voyages, notamment avec la liste limitée de voyages. De plus, il s'est occupé de la mise en place de la page « trip.php » qui permet d'afficher une vue détaillée d'un voyage lorsqu'un utilisateur clique sur un voyage spécifique. Cela est fait avec l'intégration des liens dynamiques entre « search.php » et « trip.php », ce qui assure une navigation fluide entre la liste des voyages et leur page de détails. De plus, il s'est occupé de la mise en place du processus de paiement avec CY Bank.

-Un bouton « Payer maintenant » dans « trip.php » a été ajouté. En cliquant dessus, l'utilisateur se retrouve dans la page « recap.php », page sur laquelle le montant à payer est affiché ainsi qu'un bouton permettant d'accéder au paiement via CY Bank.

-Envoi sécurisé des données de paiement à CY Bank (montant, identifiant de transaction, valeur de contrôle).

Il y a aussi une gestion du retour de paiement via « return\_payment.php » :

-Affichage du message « Paiement accepté » ou « Paiement refusé » après la validation de CY Bank. Un lien permettant de revenir sur le site dans la page « search.php »

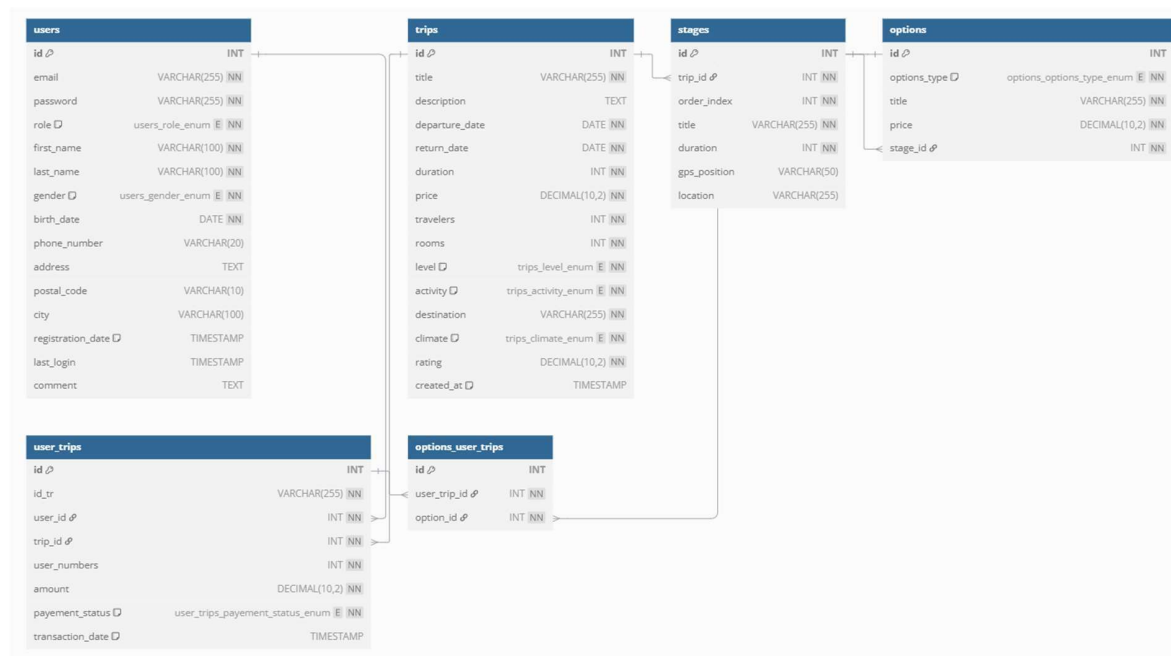
Remi et Ahmed ont collaboré pour implémenter une barre de recherche dynamique permettant aux utilisateurs de rechercher un voyage à partir de mots-clés. Un système de recherche filtrant les voyages en fonction des mots-clés saisis. Tant que la recherche à la même prononciation sonore que le terme présent dans les voyages, celle-ci fonctionnera, de même pour les fautes d'orthographe. Après la saisi, le(s) résultat(s) de recherche(s) est/sont affiché(s) sous forme de « cartes

interactives ». Une optimisation des performances a été faite en limitant les résultats affichés et en optimisant les requêtes SQL.

-Mise en place d'un système de pagination dans « search.php » : en bas de page, des boutons permettent d'afficher les voyages par groupe de 5 sans quitter « search.php ». Les 15 nouveaux voyages ont donc été ajoutés dans la base de données.

L'affichage est optimisé de manière à afficher la liste de voyage avec clarté.

### Schéma de la base de données utilisées :



### Phase 3 :

Cette phase 3 du projet a permis l'amélioration de l'interface utilisateur avec l'utilisation du Javascript. Notre objectif a été de rendre le site plus interactif, fluide et personnalisable pour l'utilisateur, tout en respectant les critères de performance, d'accessibilité et de confort de navigation.

Ahmed s'est occupé de la gestion des modes d'affichage en implémentant un bouton « mode accessible » permettant de basculer le site dans un thème plus sombre. Un fichier CSS supplémentaire a été ajouté pour ce mode, la modification dynamique du style CSS via javascript permet donc de passer en « mode accessible » sans rechargement de la

page. Le « mode accessible » repose donc sur un CSS distinct appliquant des couleurs « contrastées » (brun foncé : #413400, jaune #ffff00, #ffff66). Ce choix de thème est sauvegardé dans un cookie et est réactivé automatiquement au rechargement.

Il s'est aussi occupé de modifier (de manière dynamique) le profil utilisateur, dans « profil.php », chaque champ d'information est créé par défaut, mais peut-être rendu modifiable en cliquant sur un bouton à côté :

- Les champs deviennent éditables/modifiables individuellement.
- L'utilisateur peut valider ou annuler ses modifications localement
- Une fois une modification réalisée, un bouton de soumission globale s'affiche pour envoyer les modifications vers le serveur.

De plus, sur la page « search.php » (celle qui permet de rechercher des voyages par mots-clés), il y a ajouté un système de tri dynamique en JavaScript (sans rechargement de la page) selon plusieurs critères, à savoir : prix, durée, nombre d'étapes, dates... Les résultats sont triés côté client et réaffichés immédiatement.

Ensuite, Rémi s'est occupé de l'amélioration des formulaires sur les pages « login.php » et « profil.php », des vérifications ont été ajoutées. Les formulaires sont maintenant validés en JavaScript avant envoi, cela évite les requêtes inutiles. Un compteur de caractères informe en temps réel des limites sur les champs (emails, mot de passe). Un bouton en forme d'œil permet d'afficher ou de cacher les mots de passes. En cas d'erreur, des messages s'affichent dynamiquement sans rechargement de la page. Enfin, il a ajouté un système de panier, liée à la personnalisation de voyage :

- Lorsqu'un utilisateur modifie un voyage ou accède à « recap.php », ce dernier est ajouté à un panier temporaire.
- Cette structure permet à un utilisateur de garder en mémoire plusieurs voyages avant paiement
- La logique est gérée partiellement côté serveur via session.

Abdelwaheb s'est chargé d'introduire une simulation d'attente lors des interactions avec les utilisateurs sur « admin.php ». Lorsqu'un bouton est cliqué (les icônes : bannir, supprimer, accorder une réduction), il est temporairement créé pendant quelques secondes, mimant donc un appel serveur. Ce délai simulé permet de reproduire une latence réaliste avant activation ou mise à jour visuelle. Ensuite, il a mis à jour de manière dynamique le récapitulatif avec les prix. Dans « recap.php », il a mis en place un système JavaScript permettant de :

- Recalculer dynamiquement le prix total d'un voyage selon les options choisies
- Chaque modification (nombre de personnes, activités) met à jour le total en temps réel.
- Cela garantit une cohérence avec le prix final affiché lors de la validation serveur.

#### Phase 4 :

Pour cette 4<sup>ème</sup> et dernière phase du projet Click-Journey a pour objectif d'intégrer des requêtes asynchrones (AJAX) afin d'améliorer l'expérience utilisateur. Il s'agit là de continuer les améliorations des différentes phases précédentes en ajoutant une couche d'interactivité et de fluidité dans les échanges client-serveur. L'enjeu principal était ici, de rendre le site plus réactif, ergonomique, en évitant le rechargement complet des pages.

Pour se faire, Ahmed a mis en place un système permettant à l'utilisateur d'ajouter un voyage à son panier comme dans la phase 3, cependant, ici, cela se fait via une requête AJAX, qui transfère les données vers le serveur et met à jour le nombre d'articles dans le panier (qui est visible dans l'en-tête). Il s'est chargé d'implémenter la suppression automatique du contenu du panier après validation du paiement dans « recap\_user.php », ce qui assure une gestion cohérente des données post-transaction.

Dans « recap.php », il a aussi travaillé sur l'affichage dynamique du prix total :

- Lorsqu'un utilisateur modifie des options (activités, nombres de personnes), le total se met à jour automatiquement sans rechargement.

En rendant le récapitulatif dynamique, cela améliore la transparence sur le coût et renforce l'interactivité.

Sur la page « profil.php », chaque champs peut-être modifié via un bouton. Lorsqu'une modification est validée :

- Les nouvelles données sont envoyées via une requête AJAX à un script PHP traitant la mise à jour.
- Si la modification réussit, la nouvelle valeur est affichée en temps réel.

- Cela permet à l'utilisateur de modifier ses informations sans recharger toute la page.

Dans « search.php », Ahmed a amélioré la recherche en ajoutant à nouveau des requêtes AJAX permettant :

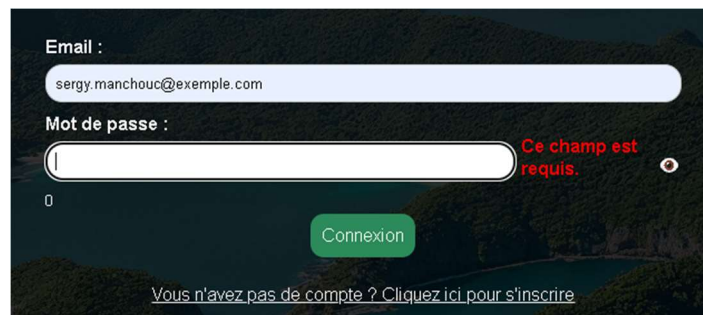
- Un filtrage dynamique des voyages (par prix, durée, destination, date).
- Une suggestion automatique de mots-clés pendant la saisie.
- Un rechargement partiel des résultats sans recharger toute la page.

Cela améliore nettement l'expérience utilisateur, notamment sur mobile.

Abdelwaheb s'est occupé de remplacer les anciennes méthodes de soumissions par des requêtes AJAX dans « login.php ».

- Lorsqu'un utilisateur tente de se connecter, les données sont envoyées à un script de vérification sans recharger la page.
- Les erreurs (identifiants invalides, champs vides, ...) sont affichées directement dans la page, sous forme de message stylisés.

Exemple :



- Même logique pour l'inscription : les champs sont validés côté client, et le retour AJAX affiche un message de confirmation ou d'erreur.

Enfin, il s'est assuré du bon fonctionnement des trois panels de redirection de la page d'accueil (« index.php ») qui orientent les utilisateurs vers des sélections de voyages spécifiques, par exemple : « stages de survie », « survie en pleine nature » ou encore « escape game de survie ». Ces panneaux envoient des requêtes AJAX avec des paramètres prédéfinis vers « search.php », permettant donc d'afficher directement les résultats correspondant à la catégorie choisie.

Rémi s'est occupé de la gestion des utilisateurs dans la page admin, sur « admin.php », il a développé des fonctionnalités asynchrones de gestion des utilisateurs :



- Lorsqu'un administrateur clique sur « bannir » ou « supprimer », la commande est envoyée en AJAX.
- La modification s'effectue côté serveur, et la ligne de l'utilisateur est mise à jour ou supprimée en direct.
- Des messages de retour confirment l'action réalisée.

Enfin, il s'est chargé de vérifier l'ensemble des fonctionnalités AJAX du projet, de déboguer les scripts JavaScript et de tester les performances (avec Ahmed et Abdelwaheb). Il a centralisé les scripts JavaScript dans des fichiers dédiés afin de faciliter leur maintenance et leur réutilisation. Il a également vérifié la cohérence des en-têtes http, le bon déroulement des appels « fetch » ou « XMLHttpRequest », ainsi que la gestion des erreurs retournés par le serveur.