

# Conception du Site

AVELINE Augustin, CRISSOT Martin, DIOP Bineta, Préing 2 - MI3 - A

## Phase 1 : html/CSS

Nom: PixelTravels

Concept: **jeux-vidéo** (trouvez des voyages en rapport avec le jeux vidéo de votre choix)

### Charte graphique:

- esthétique : type néons (fond foncé avec des touches de claire )
- Sous titre : écriture classique
- polices d'écriture : Arial, Helvetica, sans-serif
- taille de la police :
  - dans les boutons : 1em
  - au millieux des image : 4em
  - texte : 40px
  - info en minuscule : 20px
  - barre de recherche : 30px
- palette de couleur :
  - arrière plan : #261C57,
  - détail/ élément à faire ressortir : #F39639
  - titre : #aa00f9 Sur fond Foncé ET #261C57 sur fond clair
  - liens : #F205B3
  - texte : #261C57
  - Bloc ou contour pour les information : #43AED4
  - ombrage des image : #ffffff80

## Phase 2 : php

Format des données : JSON, côté php, utilisation de dictionnaires pour lire les données.

### Données :

**Utilisateur** : données stockées avec un fichier par utilisateur pour éviter les risques de perte de données lorsque deux utilisateurs différents modifient leurs données en même temps.

Cela permet une segmentation des données et permet d'éviter de devoir modifier tout le fichier utilisateurs.json à chaque modification d'un unique utilisateur.

Les fichiers ont pour nom l'adresse mail de l'utilisateur formaté (ex : [bob@bob.com](mailto:bob@bob.com) → devient bob-bob-com\_n.json, où n indique qu'il existe déjà n adresse qui correspondent au début du nom de fichier bob-bob-com).

Les données utilisateurs contiennent : email, mot de passe, id (géré grâce à fichier jouant le rôle de compteur), role (admin, normal, vip, banni), des informations personnelles, la liste des voyages achetés / consultés, des informations de connexion et des informations de

modification si un admin a changé leur statut. Elles contiennent de plus les données voyages remplies par l'utilisateur. (-> fini phase 3). Les mots de passe ne sont pas hachés pour le moment pour simplifier la maintenance / la correction.

En ce qui concerne les sessions, l'id de session est généré aléatoirement et est stocké comme cookie d'id de session.

**Voyages** : Les données de tous les voyages sont stockées dans un seul fichier .json par simplicité.

Elles contiennent : id du voyage, titre, description, genre, thème, localisation (seulement ville et pays par simplicité), la note, les dates importantes, les étapes

Les étapes contiennent : nom de l'étape, dates clés, les personnes supplémentaires, la position gps (non utilisée), et les options supplémentaires. Il y a enfin une liste des personnes ayant fait une inscription et la taille de leur groupe, le nombre de place, le nombre de place restante, le prix total et le nom de quelques images.

La recherche de voyage par mots clés se fait par à l'aide d'un filtrage de la requête, puis d'un système de score en fonction de où les mots clés sont trouvés et combien de fois (grâce à une fonction récursive).

**Paiement** : Chez l'utilisateur, un voyage à trois statuts : consulté, en cours d'achat, ou acheté. Quand l'utilisateur est redirigé vers la page de paiement, le voyage passe dans la catégorie en cours d'achat avec l'identifiant de transaction. La transaction est stockée dans transaction\_en\_cour.json. Après le paiement, l'identifiant est récupéré par le site (avec \$\_GET). Grâce à celui-ci, les informations de transactions sont chargées, stockées et transférées dans les transactions finies et le voyage passe en acheté.

## Phase 3 : JS / DOM

Un modèle de structure html a été fait pour automatiser la vérification des formulaires en js. Nous utilisons des variables css pour pouvoir modifier d'un seul coup plus facilement les couleurs du document.