Groupie Tracker - 25/26

Contexte du projet

Le projet **Groupie Tracker – Nouvelle Génération** consiste à développer une application web complète permettant de visualiser, filtrer et explorer les données d'une API réelle centrée sur des artistes et leurs concerts.

L'application doit être robuste, claire et agréable à utiliser. Toute la logique applicative (recherche, filtres, navigation, erreurs) doit être gérée côté serveur en Go. Le frontend se limite à du HTML et du CSS, sans JavaScript.

Objectifs pédagogiques:

Manipuler l'API HTTP avec le package **net/http.**

Récupérer et parser des données JSON provenant d'une API externe.

Organiser un projet Go en plusieurs packages (handlers, modèles, API, etc.).

Générer des pages web dynamiques avec **html/template.**

Gérer proprement les erreurs côté serveur.

Concevoir une interface simple, lisible et cohérente.

API Groupie Trackers

L'API officielle utilisée dans ce projet est : https://groupietrackers.herokuapp.com/api

Tu dois impérativement consommer cette API depuis ton serveur Go. Aucune base de données ni fichier JSON local ne doit être utilisé comme source principale des données.

Les Endpoints disponibles

Endpoint	Description
/artists	Liste des artistes (nom, image, année de création, premier album, membres, etc.).

/locations	Lieux des concerts des artistes.
/dates	Dates des concerts des artistes.
/relation	Lien complet artiste ↔ dates ↔ lieux.

3. Fonctionnalités obligatoires

Ton site doit impérativement proposer les fonctionnalités suivantes :

1) Page d'accueil

- Présentation de l'application.
- Navigation claire vers la liste des artistes.

2) Liste des artistes

- Affichage de tous les artistes (cartes, blocs ou tableaux, etc.).
- Affichage au minimum : image, nom, année de création, nombre de membres.
- Lien vers la page détaillée de chaque artiste.

3) Page de détails d'un artiste

- Image, nom, année de création, premier album, membres.
- Liste des concerts : dates + lieux.
- Navigation fluide (retour, autres pages).

4) Barre de recherche

- Champ de recherche basé sur une requête HTTP
- Recherche par nom d'artiste, membre, etc.
- Système de suggestion pour la barre de recherche en JS.

5) Filtres

- Au moins un filtre par intervalle (ex : année de création min/max).
- Au moins un filtre par sélection multiple (ex : nombre de membres, lieux de concert).

- Combinaison possible des filtres.

6) Carte interactive

- Pour voir les lieux et dates de concert (celle que l'on vous donnes)

7) Événement interactif

- Une action utilisateur (clic sur lien ou bouton) doit déclencher une nouvelle requête vers le serveur.
- Exemple : clic sur un lieu → page listant les concerts à ce lieu.

8) Gestion d' erreurs

- Pages d'erreur personnalisées (404, erreurs de paramètres, etc.).
- Aucun crash serveur..
- Erreurs gérées proprement dans le code Go (pas de blocs monolithiques)

Bonus possibles:

Carte: vous pouvez y placer des pop ups etc…

Cache API : mise en cache en mémoire des données avec rafraîchissement ;

Favoris : marquage d'artistes favoris via cookies ;

Comparaison : page permettant de comparer deux artistes ;

Thème sombre : thème alternatif.

Un ou deux bonus bien terminés valent mieux que plusieurs bonus incomplets.

5. Livrables

Code source complet du projet en un fichier zip.

README.md impliquant:

- L’objectif du projet,
- Comment lancer le projet,
- Les routes principales et leur fonctions,
- Les fonctionnalités implémentées (obligatoires + bonus).

Ainsi que votre gestion de projets

NB: mettre lien GitHub si vous n’avez pas utilisé gitea.

Prends le temps de concevoir ton architecture, teste régulièrement ton serveur, et soigne ton interface utilisateur.

Bonne chance avec **Groupie Tracker la team** !