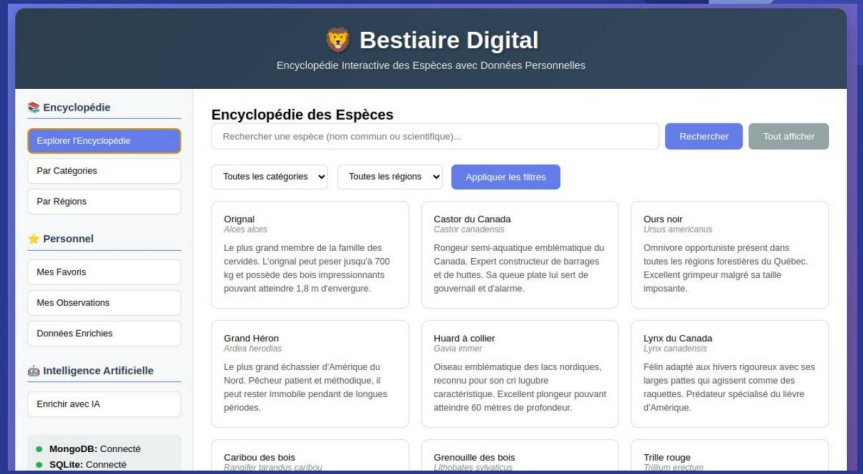
PROJET BESTIAIRE



Bestiaire Digital

Encyclopédie Interactive des Espèces avec Données Personnelles

Encyclopédie

- Explorer l'Encyclopédie
- Par Catégories
- Par Régions

Personnel

- Mes Favoris
- Mes Observations
- Données Enrichies

Intelligence Artificielle

- Enrichir avec IA

- **MongoDB:** Connecté
- **SQLite:** Connecté

Encyclopédie des Espèces

Rechercher une espèce (nom commun ou scientifique)...

Rechercher | Tout afficher

Toutes les catégories | Toutes les régions | Appliquer les filtres

Orignal
Alces alces

Le plus grand membre de la famille des cervidés. L'orignal peut peser jusqu'à 700 kg et possède des bois impressionnants pouvant atteindre 1,8 m d'envergure.

Castor du Canada
Castor canadensis

Rongeur semi-aquatique emblématique du Canada. Expert constructeur de barrages et de huttes. Sa queue plate lui sert de gouvernail et d'alarme.

Ours noir
Ursus americanus

Omnivore opportuniste présent dans toutes les régions forestières du Québec. Excellent grimpeur malgré sa taille imposante.

Grand Héron
Ardea herodias

Le plus grand échassier d'Amérique du Nord. Pêcheur patient et méthodique, il peut rester immobile pendant de longues périodes.

Huard à collier
Gavia immer

Oiseau emblématique des lacs nordiques, reconnu pour son cri lugubre caractéristique. Excellent plongeur pouvant atteindre 60 mètres de profondeur.

Lynx du Canada
Lynx canadensis

Félin adapté aux hivers rigoureux avec ses larges pattes qui agissent comme des raquettes. Prédateur spécialisé du lièvre d'Amérique.

Caribou des bois
Rangifer tarandus caribou

Grenouille des bois
Lithobates sylvaticus

Trille rouge
Trillium erectum

PROJET BESTIAIRE

Vous allez créer une APP qui présente les données d'un BESTIAIRE d'espèces enregistrées dans une base de données NoSQL distante.

Cette APP permettra de consulter les espèces, mais aussi d'interagir avec chacune pour y ajouter des informations personnelles connue de toi seul, telles que des étoiles, des mots-clés ou des notes (textes) personnelles.

Enfin, une connexion à une API intelligente permettra de l'enrichir avec de nouvelles descriptions ou de nouvelles espèces.

VERSIONNÉ dans GITHUB

LABORATOIRES EN CLASSE

Le Projet Bestiaire est une série de laboratoires réalisés en classe, dans les laboratoires, et non à la maison.

Une absence à un laboratoire entraîne la note de zéro pour ce laboratoire, sans possibilité de reprise.

Exception : en cas de maladie, une reprise est possible, pour un seul laboratoire.

PRÉSENCE OBLIGATOIRE

Étapes du projet (5 points chacune)

1 - Données NoSQL = MongoCloud

2 - Accesseurs NoSQL en POC = bestiaire-poc/test-fleurs.js

3 - Intégration à une APP JS = bestiaire.zip

4 - Données locales = Accesseur LocalStorage

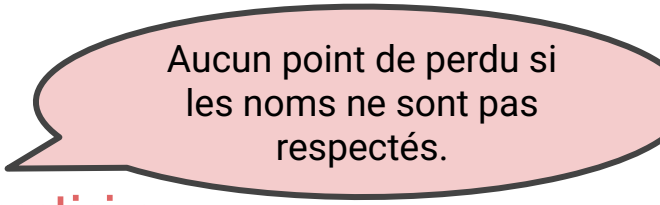
5 - API AI = Données de chat gpt api ou autre

1 - Données NoSQL = MongoCloud

Vous allez créer une collection de données de votre cru dans une base de données appelées **bestiaire** sur MongoCloud.

Étapes :

- Créer un compte MongoDB Atlas (gratuit)
- Créer un cluster ClusterDuMusee et une base de données bestiaire
- Récupérer la chaîne de connexion et ajouter 0.0.0.0/0 dans les IP permises
- Créer une collection de votre choix (équivalent de table)
- Insérer 5 documents (équivalent de objet ou enregistrement) avec 5 champs en moyenne



REMISE du SCREENSHOT dans Discord

2 - Accesseurs NoSQL en POC

Vous allez programmer des fonctions test pour manipuler les données de votre collection. Nous travaillons dans une Proof Of Concept (preuve de concept) qui n'est pas la vraie app.

Étapes du tutoriel :

https://donnees.projet.autos/bestiaire/tutoriel-2-commandes-importantes.html

- Installer Node.JS, version 22 LTS, par apt (dépôt officiel NodeSource)
- Installer le paquet mongodb
- Récupérer la chaîne de connexion et ajouter 0.0.0.0/0 dans les IP permises
- Préparer une fonction qui peut LIRE et retourner les données (appeler et faire un log)
- Préparer une fonction qui peut AJOUTER une donnée (et tester)

TAG : POC_NOSQL

```
computer$ node test-fleur.js
```

3 - Intégration à une APP JS = bestiaire.zip

Vous allez afficher dans une APP préprogrammée les données d'une nouvelle collection appelée especes . Ceci implique deux programmation distinctes : le téléchargement des données dans BestiaireDAO.js et l'affichage de ces données dans index.html. La technologie Electron permet de faire le lien entre les deux.

Etapes du tutoriel:
https://donnees.projet.autos/bestiaire/tutoriel-3-application-bestiaire.html

- Tester l'affichage de boites de données factices dans l'APP avec des document.createElement()
- Configurer config.js pour la connexion et lancer le script populate-bestiaire *(npm run populate)*
- Programmer la lecture des especes dans AccesseurBestiaire.js et l'appeler dans index.html
- Boucler dans les données et intégrer leur affichage aux jolies boites déjà décorées

TAG : LIVRAISON_APP

```
computer$ npm run start
```

4 - Données locales = Accesseur LocalStorage

Vous allez programmer la navigation vers la page détail de chaque espèce (le template html est fourni).
Dans la page détail, une donnée personnalisée est éditable par l'utilisateur et persiste localement. Cette donnée se réaffiche dans la page espèce lorsqu'on ouvre l'app à nouveau.
Par exemple des étoiles de notation, des favoris pour marquer nos espèces préférées, des notes d'observations, des mots-clés, etc. Une seule donnée affichée est requise pour réussir l'atelier.

Étapes du tutoriel:

- 1. Créer une navigation vers une page de détail pour chaque espèce
- 2. Permettre l'édition de données personnalisées (notes, étoiles, favoris, mots-clés)
- 3. Persister ces données en localStorage
- 4. Afficher ces données personnalisées sur les cartes d'espèces

Point bonus : réafficher les favoris ou mes notes persos dans la page séparée

TAG : LIVRAISON_COMPOSITE

5 - API AI = Données de Gemini - chat gpt api ou autre

Vous allez vous connecter à une API intelligente. Votre logiciel va générer un prompt basé sur le contexte de navigation (relié à l'espèce dont la page est affichée). Les données générées par l'API intelligente seront récupérées et enregistrées localement et dans le cloud.

Étapes du tutoriel:

- génére ta clé https://aistudio.google.com/api-keys
- intègre le html de la page intelligente
- teste la réponse de l'API intelligente
- personnalise le prompt différemment pour les deux sections
- enregistre les nouveaux animaux dans le cloud et les enrichissements localement

TAG : LIVRAISON_AI

6 - POINT BONUS possibles

Après la remise initiale de l'atelier tel que mis en pratique en laboratoire, on peut intégrer des variations :

- Intégration d'un visuel au choix (voir page suivante)
- Intégration d'une BDD locale autre que localstorage
- Ajout d'une seconde donnée à l'APP avec tous les éléments (page navigable, enregistrement mongo, cache locale)
- Modifier la création des boites pour utiliser un mini-template HTML
- Réafficher les favoris et les notes persos

Date limite pour les points bonus : jour de la fin de la session

/home/n/DEMO/bestiaire/template/

Split



style-moderne.html



style-grimoire.html



style-matrix.html



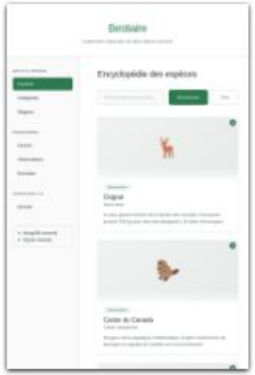
style-necromancien.html



style-nature.html



style-vintage.html



style-minimal.html



style-cabinet.html

8 Files (176.1 KiB)

Zoom:

38.6 GiB free