OK tout ça c'est à faire sans doute... mais déjà la mettre à niveau fonctionnellement avec les possibilités dont nous disposons...

0 – Interface

0.0 Faire un .bat ( ?) pour ouvrir la console  
Contrôle d’accès et Abbréviations :  
V= Visualisation est omis : elle est général sur l’application   
CMS = Création, Modifications, Suppression logique par drapeau  
CM\*S\* = Création , Modifications autres que l’identifiant logique**(code, adresse\_mail ou nom\_dossier**), Suppression physique   
donc  
CM\*S = Création , Modifications autres que l’identifiant logique(code ou adresse mail par exemple), Suppression logique par drapeau  
A ce stade pas de développement pour changer un « drapeau » : Admin en base…

0.1 Présentation (CSS) :  
 Avoir les deux présentations du Cockpit :  
sobrePro.css : se rapprocher de Github (style, icônes etc…)   
pas de tour au bouton ou très léger gris clair, icône genre Electron ou Github moins enfantine…  
cockpit.css : tu restes maitre d’ouvrage, d’œuvre !

0.2 – Interface découpage :  
Etat : 3 colonnes fixes et un commentaire affiché à ne pas afficher(// ✅ pilotHub/index.html)  
On les appelle : ColA, ColB, ColC

Modifications de présentation :  
Ajouter un bandeau sur toute la longueur divisé en trois :   
a gauche : icône pour effacer/afficher la ColA avec le message contextuel « afficher/cacher « Domaines »)  
au centre  l’environnement : Dev ou Prod avec l’alarme   
à droite une icône que je dois te fournir

Rendre les colonnes « resizable »(retaillable)

Modifications de contenu :  
ColA :   
Titre : «1 -  Domaine et Dossiers des Questionnaire »(à gauche et une icône à droite(fonction « Ajouter Domaine »avec message contextuel et contrôle rôle :Admin)   
liste indenté des Domaines/Questionnaires (Accordéon si pas contraignant ??)

Domaine : sur clic sur nom = affichage des documents des types acceptés placés dans le dossier « docs » du répertoire  
Questionnaires : sur clic sur nom = affichage des documents des types acceptés placés dans le dossier « docs » du répertoire, après de nom avoir une icône avec message contextuel «Maintenance » 🡺 sur clic(CM\*S en ColB avec contrôle rôle :Admin et Auteur et M\* sur contrôle en table) – Détail voir   
  
Ligne de séparation  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Titre : « 2 - Collectif »  
Sous-Titre : « 2.1 - Groupe » 🡺 sur clic(CM\*S en ColB avec contrôle rôle :Admin)  
Sous-Titre : « 2.1 - Collaborateurs » 🡺 sur clic(CM\*S en ColB avec contrôle rôle :Admin)  
Ligne de séparation  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Titre : « 3 – Types des documents » 🡺 sur clic(CM\*S en ColB avec contrôle rôle :Admin)

**1 - Pouvoir ouvrir l'interface sans passer par Powershell (par ex;) ==> fonction cockpi**

Pour mémoire (Powershell) :  
> cd C:\serveurNodeJS\pilotHub  
> npm run dev  
  
Rappel : 0.0 Faire un .bat ( ?) pour ouvrir la console

2 - lier l'affichage aux sources de données, le maintenir avec ce que l'on a... et suivant quelques choix

Tu as créé des fichiers pour générer les tables de la base qui sont à un niveau correct et appelé la table des domaines d’application « domaine » donc on doit avoir par questionnaire une caractéristique que l’on avait appelé « domain »e et s’appellera du coup « genre » de questionnaire (ex : sondage ou casserole)  
Convention : en table les rubriques libellées **code, adresse\_mail ou nom\_dossier ne sont pas modifiables**  
Tables à créer en plus :  
**Table** « **Genre** » : id, nom NOT NULL UNIQUE, description, statut DEFAULT true - Et initialisée avec (.Aucun, Pas défini, true)   
**Table** « **Docs** » : id, code NOT NULL UNIQUE, statut DEFAULT true - Et initialisée avec (.html, true) ;(.docx, true)   
**Table** des **rôles** pas modifiable pour l’instant : id, nom NOT NULL UNIQUE, description, statut DEFAULT true - Et initialisée avec (admin, tous les droits sur l’application, true) ; (auteur, tous les droits sur le questionnaire, true) ; (utilisateur, droit (M) sur la fonction, true)\* ; (visiteur, voir l’application, true) ;  
**Table** à modifier :  
**questionnaires :** change proprietaire en auteur et ajouter genre\_id INTEGER,  
  
\* c’est un problème de développement important : pour l’instant le faire avec (M) sur l application donc (pas de supp ni de création uniquement sur l’ensemble et pas d’affectation par questionnaire) à affiner plus tard…

2.1 C'est les données en "base" qui pilote le système et qui le font suivre:

* Définition de la structure au niveau 1 (des domaines d’application) et niveau 2 (questionnaires)  
  Fonctions de :  
   création : en base et dans et par le FS (File System)  
   MàJ(M\*): en base +🡺chemin dynamique malgré tout (sur id)!  
  **suppression(S) : en base avec un flag (le FS est maintenu par l’ « Useur »)**

**3 –Interfaces et fonctionnalités**

**Les interfaces d’actions sur** colA sont affichés dans ColB  
Si possible un seul module pour les tables « standard :  
sous forme de liste en tableau :

|  |  |
| --- | --- |
| Nom de rubrique en table débarrassée des « \_ » remplacé par «  »(1 blanc) | Valeur avec contôle en table format, nécessité… |

Bouton « save »  
Si on peut dans la table avoir (c’est que les base de données se sont améliorés ;-)) quelque chose comme çà pour un champ/rubrique:  
domaine\_id INTEGER **REFERENCE** ‘**domaine’**, ou sinon contrôler dans la database ou sinon faire une table des dépendances et afficher une combo « moderne » pour choisir la valeur choisie   
PS : faire attention avec les rubriques les noms au pluriel et au singulier (tout au singulier c’est mieux et générique) **:Trouver un compromis  
  
Le faire pour les tables**

**domaine, genre, doc(s), utilisateur(s), questionnaire(s)**

**En réserve groupe:**