Traitement d'un fichier json avec interface en Qt

Données proposées (mais non imposée : toute autre proposition doit être validée par l'enseignant).

- ⋆ Dans le dépôt : https://github.com/manami-project/anime-offline-database
  Télécharger le fichier `anime-offline-database.json` et vérifier la licence.
- ⋆ Dans le dépôt : https://github.com/Purukitto/pokemon-data.json
  Télécharger le fichier `pokedex.json`. Attention à la licence !

Sélection — 6

En tant qu'utilisateur, je souhaite pouvoir rechercher une donnée dans le fichier, sans avoir à le parcourir à la main.

- ⋆ Un champ de recherche par mot-clé est proposé.
- ⋆ Les résultats possibles s'affichent dans une liste.
- ⋆ Le résultat souhaité peut être sélectionné.

Sauvegarde — 5

En tant qu'utilisateur, je souhaite que mes actions dans l'application soient sauvegardées, pour être poursuivies d'une utilisation à l'autre.

- ⋆ Si besoin : possibilité de chargement.
- ⋆ Possibilité de sauvegarde (voire automatique).

Image — 4

En tant que marketing, je souhaite que certaines informations soient visuelles, afin que l'application ne soit pas austère.

- ⋆ Au moins une information apparait sous la forme d'une image.

Mode authentifié — 5

En tant qu'administrateur, je souhaite pouvoir faire des actions supplémentaires par rapport aux autres utilisateurs, pour maintenir l'application à jour.

- ⋆ L'administrateur peut s'identifier (mot de passe).
- ⋆ L'administrateur a accès à plus de fonctionnalités.

json
(format standard pour un fichier de données représenté en Python par une liste de dictionnaires.)

On utilise la bibliothèque `json` :

```python
import json

with open('anime-offline-database.json', encoding="utf8") as f:
    contenu = json.load(f)

print(contenu['data'][24235]['picture'])

with open('perso.json', 'w', encoding='utf-8') as f:
    json.dump(contenu['data'][200:210], f, ensure_ascii=False, indent=4)
```

Qt (Pyside6 : Qt for Python)
(Framework de développement d'application multi-plateforme.)

Aide concernant le chargement d'une image désignée par une URL :

```python
class Label_image(QLabel):

    def __init__(self, url:str, parent=None):
        super().__init__(parent)
        self.manager = QNetworkAccessManager()
        self.manager.finished.connect(self.chargement_fini)
        url = QUrl.fromUserInput(url)
        self.manager.get(QNetworkRequest(url))

    @Slot(QNetworkReply)
    def chargement_fini(self, reponse:QNetworkReply):
        image = QImage()
        image.loadFromData(reponse.readAll())
        pixmap = QPixmap.fromImage(image)
        self.setPixmap(pixmap)
```