Soutien AI :

Alek :

# Question 1

Avec Python pyside6 qt, peux tu me faire une page avec trois boutton sur la gauche nommé : ouvrir, modifier, suprimer; sur la droite une liste qui contient 4 colonne nommé : nom,prenom, username, poste; et puis me faire un exmple d'incertion dans la liste

## réponse

import sys

from PySide6.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout, QPushButton, QTableWidget, QTableWidgetItem, QHeaderView

from PySide6.QtCore import Qt

class MainWindow(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

# Initialisation de la fenêtre principale

self.setWindowTitle("Gestion des Employés")

self.setGeometry(100, 100, 800, 400)

# Layout principal

main\_layout = QHBoxLayout()

# Partie gauche (les boutons)

self.sidebar = QVBoxLayout()

# Boutons à gauche

self.ouvrir\_button = QPushButton("Ouvrir")

self.modifier\_button = QPushButton("Modifier")

self.supprimer\_button = QPushButton("Supprimer")

# Ajout des boutons dans la sidebar

self.sidebar.addWidget(self.ouvrir\_button)

self.sidebar.addWidget(self.modifier\_button)

self.sidebar.addWidget(self.supprimer\_button)

# Partie droite (la table)

self.table = QTableWidget()

# Initialisation de la table avec 4 colonnes

self.table.setColumnCount(4)

self.table.setHorizontalHeaderLabels(["Nom", "Prénom", "Username", "Poste"])

# Autres configurations de la table

self.table.horizontalHeader().setSectionResizeMode(QHeaderView.Stretch)

# Ajouter des données initiales

self.insert\_initial\_data()

# Ajout des éléments à la fenêtre principale

main\_layout.addLayout(self.sidebar)

main\_layout.addWidget(self.table)

self.setLayout(main\_layout)

def insert\_initial\_data(self):

# Exemple d'insertion dans la table

data = [

("Dupont", "Jean", "jdupont", "Développeur"),

("Martin", "Sophie", "smartin", "Responsable"),

("Lemoine", "Paul", "plemoine", "Analyste"),

("Durand", "Claire", "cdurand", "Directeur")

]

# Insertion des données dans la table

for row\_idx, row\_data in enumerate(data):

self.table.insertRow(row\_idx)

for col\_idx, value in enumerate(row\_data):

self.table.setItem(row\_idx, col\_idx, QTableWidgetItem(value))

# Lancer l'application

if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication(sys.argv)

window = MainWindow()

window.show()

sys.exit(app.exec())

## soutien:

bon blueprint de page pour commencer ma liste avec des exemple

# Question 2

J’aimerais une fonction qui me permettrait d’aller chercher le nom de mes colonnes dans ma base de données :

## Réponse

db\_manager = DatabaseManager('erp\_database.db')

column\_query = "PRAGMA table\_info(Succursales);"

columns\_info = db\_manager.execute\_query(column\_query)

## soutien :

utile pour demander toutes les info d’une table (ajouter des champs)

# Question 3

J’aimerais une système d’historique pour les pages pour pouvoir mettre un bouton back qui reviens a la page précédente

## Réponse

Dans le constructeur : self.history = []

Quand tu change de page : self.history.append(self.stacked\_widget.currentWidget())

Quand tu veux revenir en arrière : if self.history:

previous\_widget = self.history.pop() # Retirer le dernier widget visité

self.stacked\_widget.setCurrentWidget(previous\_widget)

## soutien :

Permet de revenir en arrière, pouvoir revenir a différente page selon on vient de ou

# Question 4

J’aimerais changer une table pour lui ajouter un nouveau champ, j’ai un champ input\_nom qui est le nom du champ et un input input\_type qui contient le type

## Réponse

def ajouter(self):

# Vérifiez que les champs sont remplis

if self.input\_nom.text() != "" and self.input\_type.currentText() != "":

nom\_table = self.mode

nom\_new\_champ = self.input\_nom.text()

type\_new\_champ = self.input\_type.currentText()

db\_manager = DatabaseManager('erp\_database.db')

query = f"ALTER TABLE {nom\_table} ADD COLUMN {nom\_new\_champ} {type\_new\_champ};"

try:

# Exécution de la requête

db\_manager.execute\_update(query)

print(f"Colonne '{nom\_new\_champ}' ajoutée à la table '{nom\_table}'.")

except sqlite3.Error as e:

print(f"Erreur lors de l'ajout de la colonne : {e}")

## soutien :

Permet de faire la page ajouter champ

# Question 5

J’aimerais une fonction qui vérifie si ma table Employes est vide

## Réponse

def is\_table\_empty(self):

count = self.db\_manager.execute\_query("SELECT COUNT(\*) FROM Employes")

return count[0][0] == 0 # Retourne True si la table est vide

## soutien :

Permet de vérifier pour la première connexion si un employé existe

# Question 6

Qu’est ce qui serait une bonne manière dans mon erp de toujours savoir je suis dans quel succursale, quel est mon rôle, etc..

## Réponse

Tu devrais te faire une classe que tu peux appeler de partout pour savoir ou tu es ou qui tu es : class Emplacement:

# Attribut de classe (variable globale)

succursalesCode = 0

employeID = 0

## soutien :

Permet d’afficher les employé le stock de la succursale ou je suis

# Question 7

voici un exemple de ma base de données pour aller chercher un employé et afficher son info, j'aimerais une fonction qui va chercher tout mes employé et m'affcihe toutes leur informations : try: db\_manager = DatabaseManager('erp\_database.db') query = "SELECT \* FROM Employes WHERE id\_employe = ?" results = db\_manager.execute\_query(query, (1,)) for row in results: nom = row['nom'] prenom = row['prenom'] print(f"Nom : {nom}, Prénom : {prenom}") except sqlite3.Error as e: print(f"Une erreur est survenue : {e}")

## Réponse

try: db\_manager = DatabaseManager('erp\_database.db') # Chemin de votre base de données # Requête SQL pour récupérer tous les employés query = "SELECT nom, prenom, username, poste FROM Employes" results = db\_manager.execute\_query(query) if results: # Affichage des résultats dans la console avec print print(f"{'Nom':<20} {'Prénom':<20} {'Username':<20} {'Poste':<20}") print("-" \* 80) # Séparateur pour améliorer la lisibilité for row in results: nom, prenom, username, poste = row print(f"{nom:<20} {prenom:<20} {username:<20} {poste:<20}") else: print("Aucun employé trouvé dans la base de données.")

## soutien :

Quand je veux vérifier l’information de mes employé