

Mini projet

Interface Graphique et Python



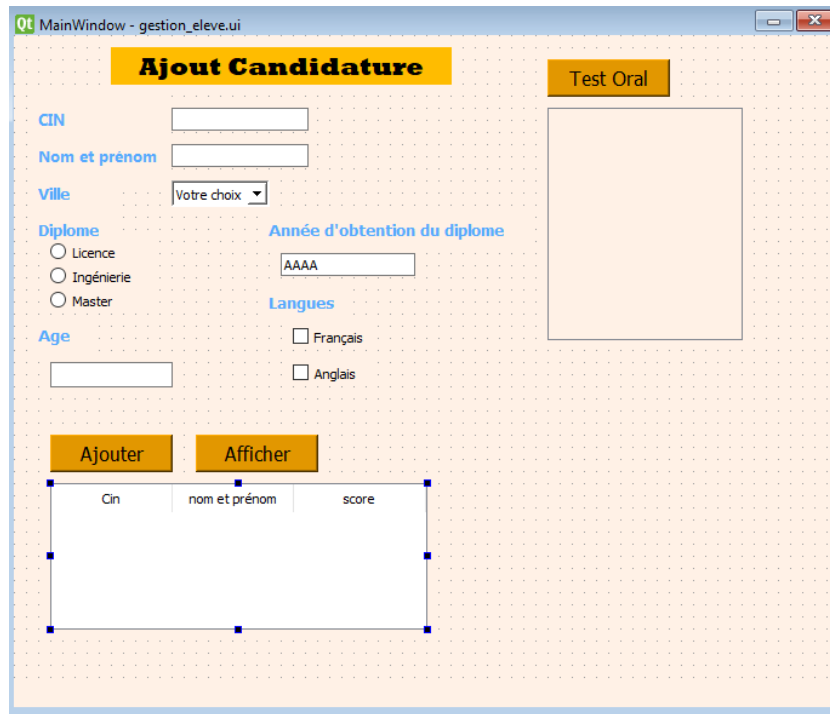
Une société désire faire des recrutements dans les postes vacants pour satisfaire ses besoins, pour cela le responsable des ressources humaines a décidé de créer une application informatique pour que les candidats puissent poser leurs candidatures.

Sachant qu'un candidat possède les informations suivantes :

- **Cin** : Chaîne de 8 chiffres, commence par 0 ou 1
- **Nom et prénom** : chaîne contenant que des lettres alphabétiques ou espace
- **Ville** : une chaîne qui désigne la ville d'un candidat
- **Diplôme** : une chaîne qui désigne le diplôme du candidat
- **Age** : entier >20
- **Année d'obtention du diplôme** : entier entre 2008 et 2023
- **Langue** : chaîne qui contiendra une ou deux langues choisies par un candidat, sinon elle contiendra le message « aucune langue »
- **Score** : entier entre 0 et 20 (initialement vide)

On se propose de concevoir une interface graphique en utilisant Qt designer contenant les éléments suivants :

- Un label pour le titre
- Un label contenant le texte «**CIN**»
- Un label contenant le texte «**Nom et prénom**»
- Un label contenant le texte «**prénom** :»
- Un label contenant le texte «**Ville**»
- Un label contenant le texte «**Diplome**»
- Un label contenant le texte «**Age**»
- Un label contenant le texte «**Année d'obtention du diplôme**»
- Un label contenant le texte «**Langues**»
- 4 zones de texte (line Edit)
- Une liste de sélection (Combo Box) contenant (votre choix, Nabeul, sousse, Tunis)
- Trois radio Bouton (Radio Button)
- Deux cases à cocher (check Box)
- Un bouton intitulé **Ajouter** permettant d'ajouter un candidat au fichier **candidat.dat**
- Un bouton intitulé **Afficher** permettant d'afficher dans la table widget les données de tous les candidats triés dans l'ordre décroissant des scores
- Un bouton intitulé **test oral** permettant d'afficher les noms des candidats qui sont invités à passer un test oral
- Une table Widget contenant les colonnes "CIN", "nom et prénom" et "score" pour afficher le contenu du fichier



Travail demandé :

- 1) Concevoir une interface graphique comme illustré ci-dessus et l'enregistrer sous le nom «AjoutC.ui »
- 2) Créer un fichier python et l'enregistrer dans votre dossier de travail sous le nom **AjoutC.py**
- 3) Développer dans votre fichier **AjoutC.py** une fonction **Score ()** qui permet de calculer le score d'un candidat comme suit :

$$\text{Score} = N1+N2+N3+N4$$

Sachant que :

- **N1 : Note1 entre 2 et 6 attribuée au candidat comme suit :**
 - N1=2 Si diplôme est licence
 - N1=4 Si diplôme est ingénierie
 - N1=6 Si diplôme est Master
- **N2 : Note2 entre 2 et 4 attribuée au candidat comme suit :**
 - N2=2 Si age<25
 - N2=4 Sinon
- **N3 : Note3 entre 3 et 6 attribuée au candidat comme suit :**
 - N3=3 Si Année<=2010
 - N3=6 Sinon
- **N4 : Note4 entre 0 et 4 attribuée au candidat comme suit :**
 - N4=0 Si aucune langue
 - N4=2 Si langue est français seulement ou anglais seulement
 - N4=4 sinon

4) Dans votre fichier AjoutC.py:

- Ajouter les instructions permettant d'appeler l'interface graphique intitulée " **AjoutC.ui** " en exploitant l'annexe ci-après.
- Développer le module **Ajout()** qui s'exécute suite à un clic sur le bouton "Ajouter", et permettant, lorsque toutes les contraintes sont respectées, d'ajouter un candidat au fichier "**candidat.dat**" sinon d'afficher, dans le cas contraire, un message d'alerte via "**QMessageBox**".
- Développer le module **AffCandidat()** dans la table widget les données de tous les candidats triés dans l'ordre décroissant des scores
- Développer le module **affTest()** permettant d'afficher les noms des candidats qui sont invités à passer un test oral dans une liste widget et les enregistrer dans un fichier texte intitulé « **testoral.txt** » sachant qu'un candidat est accepté si son score est supérieur à 10.

Annexe (Liaison de l'interface avec python)

```
from PyQt5.QtWidgets import *
from PyQt5.uic import*
...
app = QApplication([])
nom_fenetre=loadUi("Nom_interface.ui")
nom_fenetre.show()
nom_fenetre.nom_bouton.clicked.connect(module)
app.exec_()
```