**RAPPORT DU PROJET INTEGRATEUR**

PROJET INTEGRATEUR

« ORIENTATION »

*Projet réalisé par*

Moussa emhamed.

Mohamed salem

Mohamed abderahman Mohamed lemin

Taleb bahan

*Projet encadré par*

Dr. Cheikh Dhib

**REMERCIEMENTS**

Tout d’abord, nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance aux personnes suivantes, pour leur dévouement et leur soutient dans la concrétisation de ce projet ingénieur :

* Dr. Cheikh Dhib responsable projet, pour ses conseils éclairés, sa patience, sa disponibilité et pour la confiance qu’il nous a accordée dès l’ébauche du projet et tout au long de ces longues périodes , et pour nous avoir accordé toute la confiance nécessaire pour élaborer ce projet librement, et avoir mis à notre disposition tous les moyens disponibles.

1. **INTRODUCTION**

Cette application consiste à implementer un algorithme qui permet d’orienter Les étudiants dans les filières ou specialites en fonction de leurs choix et leur classement.

* Les outils de conception :
  + Draw / modulo /smartdraw

* Les outils de realisation :
  + Langage : php/python
  + Base de donne : mysql

# PROCEDURE

* On a commencé par la base du site web par écrire un programme en python qui oriente les élèves à leurs filières dépendent de leur moyenne, Ça veut dire qu'il y a une moyenne commune entre les filières , on a tiré les choix avec un ficher Excel.
  + Code:

Nbl ← le nombre de ligne de la fuel1

Pour i ← 1 a nbl-1 faire

B ← une list vide

l ← 0

Pour j ← 2 a nbl faire

Si la valeur de feul1 de colon(6),ligne(j)=i alors

d ← valeur de feul1 de colon(1),ligne(j)

Pour ← 2 a nbl faire

si valeur de feul2 de colon(1),ligne(r)=d alors

pour z ← 2 a 4 faire

Ajouter au liste(B) la valeur de feul1 de-colon(z),ligne(r)

Fin pour

Fin si

Fin pour

Pour k ← 0 a 2 faire

Si la liste(B) a la position k =”CNM” ET a#0 alors

Fuel3 de la colonne(1),ligne(i+1)← d

Fuel3 de la colonne(1),ligne(i+1)← la liste(B) a la -position(k)

a← a-1

l← 1

Sortir de boucle si la condition si et vérifier

Sinon la liste(B) a la position(k) =”RSS” ET e#0 alors

Fuel3 de la colonne(1),ligne(i+1)← d

Fuel3 de la colonne(1),ligne(i+1)← la liste(B) a la -position(k)

e← e-1

l← 1

Sortir de boucle si la condition si et vérifier

Sinon la liste(B) a la position(k) =”DSI” ET c#0 alors

Fuel3 de la colonne(1),ligne(i+1)← d

Fuel3 de la colonne(1),ligne(i+1)← la liste(B) a la -position(k)

c← c-1

l← 1

Sortir de boucle si la condition si et vérifier

Fin si

Fin pour

Fin si

Si l=1 alors

Sortir de boucle si la condition si et vérifier

Fin si

Fin pour

Fin pour

* Après on a écrit un programme en python qui oriente les étudiants par filière (chaque filière a leur propre moyenne de classement).

etudiants = {

'A': ['CNM', 'DSI', 'RSS'],

'B': ['CNM', 'DSI', 'RSS'],

'C': ['DSI', 'CNM', 'RSS'],

'D': ['DSI', 'RSS', 'CNM'],

'E': ['RSS', 'DSI', 'CNM'],

'F': ['RSS', 'CNM', 'DSI']

}

classements = {

'DSI': ['C', 'E', 'D', 'B', 'F', 'A'],

'RSS': ['B', 'F', 'A', 'E', 'D', 'C'],

'CNM': ['A', 'B', 'C', 'D', 'E', 'F']

}

places = {

'DSI': 2,

'RSS': 2,

'CNM': 1

}

def orienter\_les\_etudiants(etudiants, classements, places):

etudiants\_en\_attente = [etudiant for etudiant in etudiants]

resultats = {choix: [] for choix in places}

def obtenir\_une\_liste\_attente\_sans\_etudiant(etudiant):

return [x for x in etudiants\_en\_attente if x != etudiant]

def obtenir\_des\_resultats\_tries\_avec\_eleve(etudiant, choix):

resultats[choix].append(etudiant)

return [x for x in classements[choix] if x in resultats[choix]]

while etudiants\_en\_attente:

for etudiant in etudiants\_en\_attente.copy():

if not etudiants[etudiant]:

etudiants\_en\_attente = obtenir\_une\_liste\_attente\_sans\_etudiant(etudiant)

continue

choix = etudiants[etudiant].pop(0)

if len(resultats[choix]) < places[choix]:

resultats[choix] = obtenir\_des\_resultats\_tries\_avec\_eleve(etudiant, choix)

etudiants\_en\_attente = obtenir\_une\_liste\_attente\_sans\_etudiant(etudiant)

else:

if classements[choix].index(etudiant) < classements[choix].index(resultats[choix][-1]):

resultats[choix] = obtenir\_des\_resultats\_tries\_avec\_eleve(etudiant, choix)

etudiants\_en\_attente = obtenir\_une\_liste\_attente\_sans\_etudiant(etudiant)

etudiants\_en\_attente.append(resultats[choix].pop())

return resultats

print(orienter\_les\_etudiants(etudiants, classements, places))

* Après on a couverture se programme en Php.

<?php

$etudiants = [ 'A'=> ['CNM', 'DSI', 'RSS'],

'B'=> ['CNM', 'DSI', 'RSS'],

'C'=> ['DSI', 'CNM', 'RSS'],

'D'=> ['DSI', 'RSS', 'CNM'],

'E'=> ['RSS', 'DSI', 'CNM'],

'F'=> ['RSS', 'CNM', 'DSI']

];

$classements = ['DSI'=> ['C', 'E', 'D', 'B', 'F', 'A'],

'RSS'=> ['B', 'F', 'A', 'E', 'D', 'C'],

'CNM'=> ['A', 'B', 'C', 'D', 'E', 'F']

];

$places = ['CNM' => 1,

'DSI' => 2,

'RSS' => 2];

function obtenir\_une\_liste\_attente\_sans\_etudiant($etudiant, $etudiants\_en\_attente)

{

$liste\_attente=[];

foreach ($etudiants\_en\_attente as $s) {

if ($s != $etudiant) {

array\_push($liste\_attente ,$s );

}

}

return $liste\_attente;

}

function obtenir\_des\_resultats\_tries\_avec\_eleve($etudiant, $choix, $classements, $resultats)

{

array\_push($resultats[$choix],$etudiant);

$l=[];

foreach ($classements[$choix] as $x){

if (in\_array($x,$resultats[$choix])){

array\_push($l,$x);

}

}

return $l;

}

function orienter\_les\_etudiants($etudiants, $classements, $places){

$etudiants\_en\_attente = ["A","B","C","D","E","F"];

$resultats = ['CNM' => [],

'DSI' => [],

'RSS' => [],];

while (sizeof($etudiants\_en\_attente) > 0){

$copy\_etudiants\_en\_attente = $etudiants\_en\_attente;

foreach($copy\_etudiants\_en\_attente as $etudiant) {

if (sizeof($etudiants[$etudiant]) == 0) {

$etudiants\_en\_attente = obtenir\_une\_liste\_attente\_sans\_etudiant($etudiant,$etudiants\_en\_attente);

continue;

}

$choix = array\_shift($etudiants[$etudiant]);

if (sizeof($resultats[$choix]) < $places[$choix]) {

$resultats[$choix] = obtenir\_des\_resultats\_tries\_avec\_eleve($etudiant, $choix ,$classements,$resultats);

$etudiants\_en\_attente = obtenir\_une\_liste\_attente\_sans\_etudiant($etudiant,$etudiants\_en\_attente);

}

else{

if (array\_search($etudiant,$classements[$choix]) < array\_search($resultats[$choix][sizeof($resultats[$choix])-1] , $classements[$choix])){

$resultats[$choix] = obtenir\_des\_resultats\_tries\_avec\_eleve($etudiant, $choix,$classements,$resultats);

$etudiants\_en\_attente = obtenir\_une\_liste\_attente\_sans\_etudiant($etudiant,$etudiants\_en\_attente);

array\_push($etudiants\_en\_attente,array\_pop($resultats[$choix]));

}

}

}

}

return $resultats;

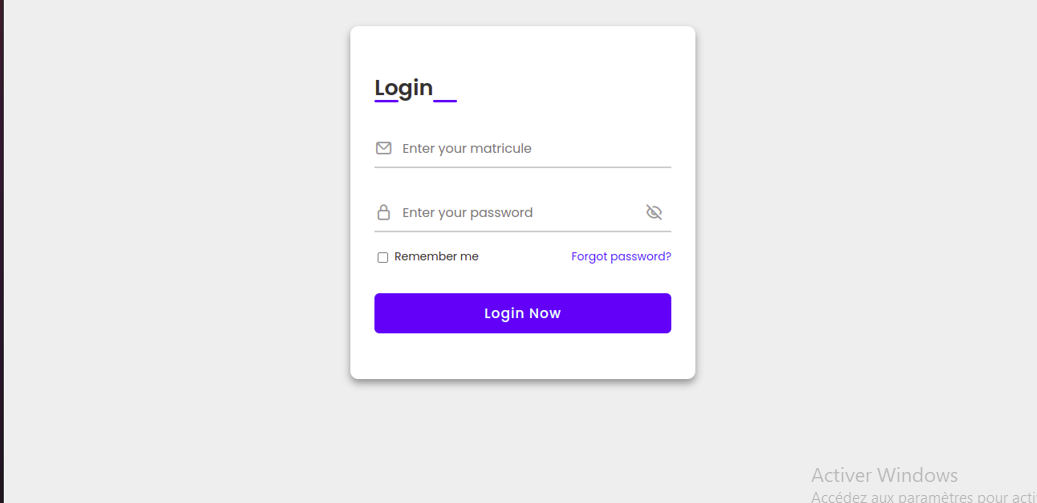
}

$k = orienter\_les\_etudiants($etudiants, $classements, $places);

print\_r($k);

?>  
plt.hist(l)  
plt.show()

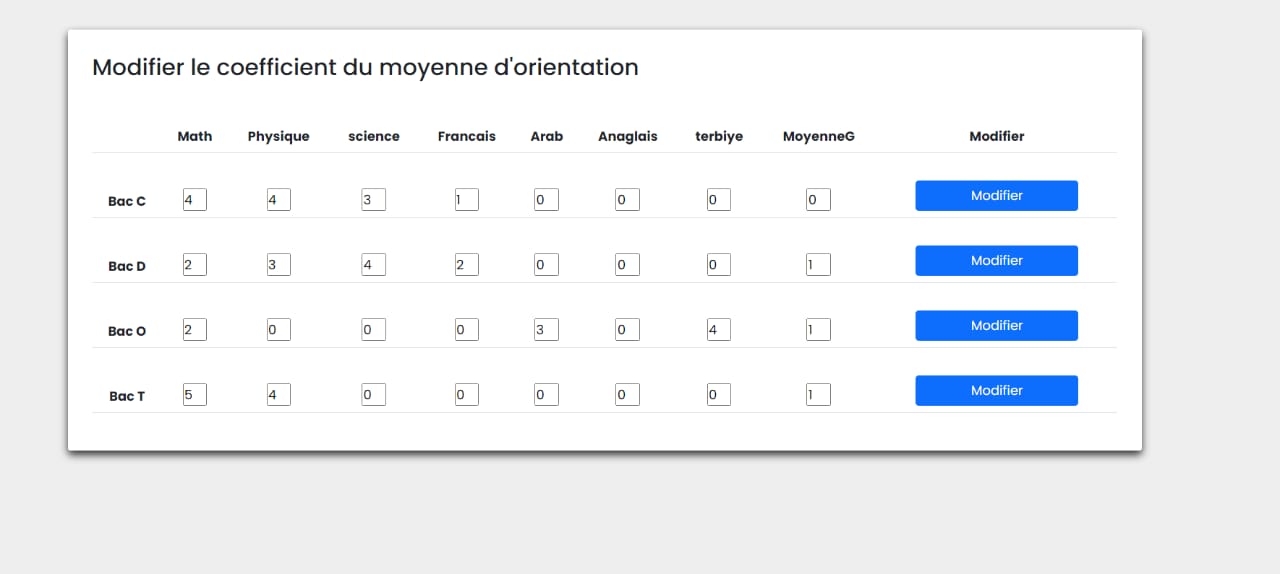
* Le site contient :
  + une page pour login qui est basée sur un code de Php qui vérifie l'authentification de l’étudiant ou l’administrateur.

 ;

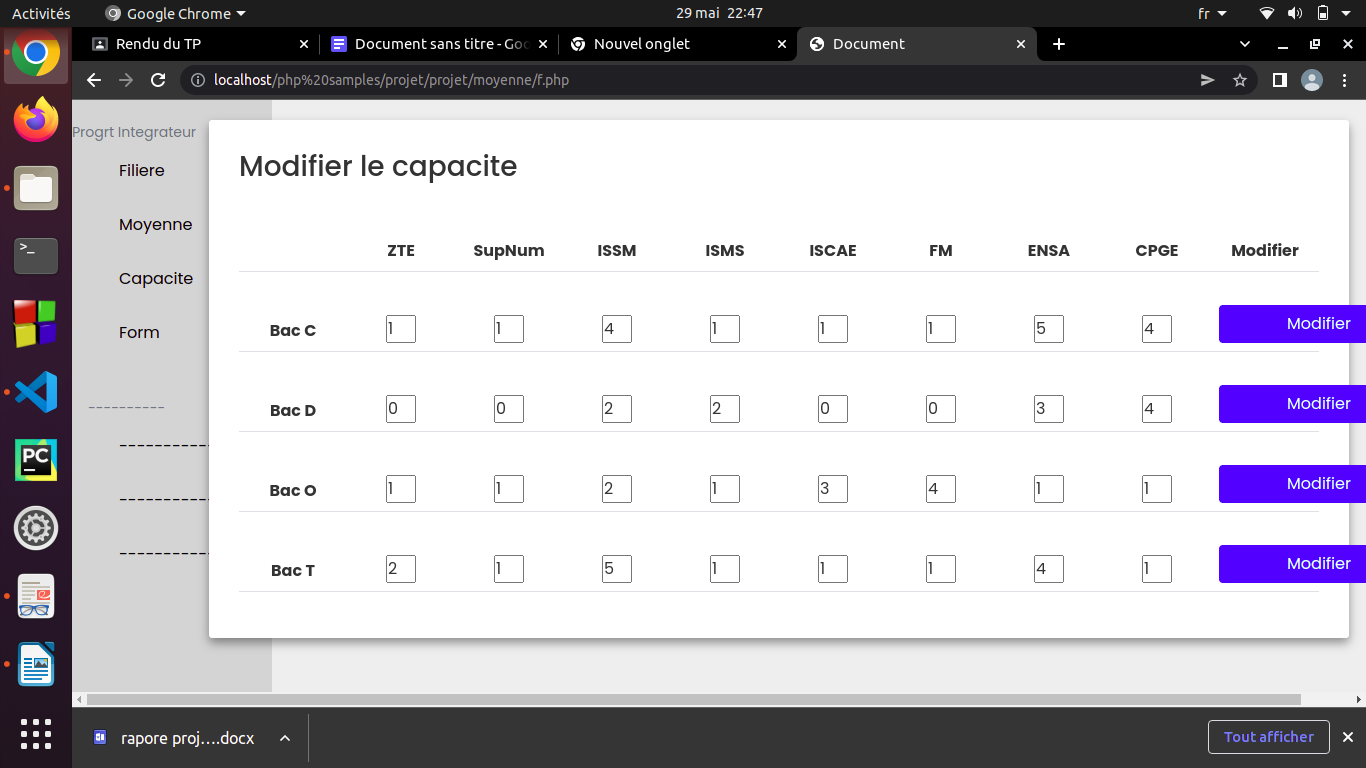
* + une page pour l’étudiant d’où il va trouver le formulaire à remplir selon leur filière.
  + une page pour l’administrateur pour qu’il puisse ajouter ou modifier ou supprimer une filière .



* + une page pour l’administrateur pour ( lieu à la page précédente ) pour qu’il puisse rentrer les coefficients des matières d’une filière .



* + une page pour l’administrateur pour qu’il puisse ajouter les capacités des filières .



1. **Implimentation**

* Les outils de conception :
  + Draw / modulo /smartdraw

* Les outils de realisation :
  + Langage : php/python
  + Base de donne : mysql