*MonAvenir*, vaste projet !

Objectif : Réaliser un mini Parcoursup, une application web *MonAvenir* destinée à recueillir et gérer les vœux d'affectation des futurs étudiants de l'enseignement supérieur dans le Rhône.

Etape 0 : En amont

* Présentation rapide du fonctionnement de Parcoursup.
* Présentation du projet de BD *MonAvenir* qui tiendra lieu de fil rouge pour cette partie du programme.
* Cahier des charges de l’application - Identification des besoins : formulaires de saisie et de consultation des vœux, base de données pour traiter les candidatures, interactions client / serveur …

D’où l’intérêt des SGBD.

* Cours sur les BD, utilisation de SQLITE Browser puis SQLITE3 de python

Nous proposons de travailler par ateliers.

En fonction du niveau des élèves, ils pourront faire tout ou partie des ateliers.

Il sera aussi possible de répartir le travail entre les élèves qui auront l’objectif commun d’achever ce projet.

Etape 1 : Requêtes SQL

* Discussion avec le groupe classe : notion de dépendances fonctionnelles, identification des entités et relations en jeu dans la BD *Monavenir* et choix du schéma de conception.
* Ecrire des requêtes SQL d’interrogation de la DB *MonAvenir*.
* Ecrire les requêtes SQL permettant la construction de la BD *MonAvenir*.
* Enoncer les fonctionnalités attendues de l’application *MonAvenir*.

Les élèves commencent à s’approprier la BD *MonAvenir*, ce qui facilitera le travail par la suite.

Etape 2 : Données open data et Fake

Construire les fichiers source au format CSV

* Création de fichiers csv pour les lycées et les écoles supérieures avec des données open data.
* Création de fichiers csv pour les élèves et les candidatures avec des données fictives

Cette étape permet de faire des révisions de première sur le traitement des données en tables, l’utilisation de fichiers CSV

Nous fournissons du code à comprendre puis des fonctions à compléter ou à écrire en utilisant l’existant.

Les tâches sont de difficultés variées ce qui permet de gérer l’hétérogénéité entre les élèves.

Etape 3 : Python et fichiers CSV

Peuplement de la BD MonAvenir à partir des fichiers CSV précédents

Etape 4 : Développement WEB côté serveur en Python, Flask

Formulaires de connexion à la BD et de recherche dans la BD avec leurs traitement dans le fichier main

Etape 5 : Traitement des candidatures Non développé. En prolongement. Le code est fourni.

Etape 6 : Création des formulaires de consultation des résultats et leurs traitements – TP noté par groupes

Etape 7 : CSS en bonus

Etape 1 : Requêtes SQL

**Atelier 1 : Requêtes SQL de consultation de la BD MonAvenir**

Quelques requêtes possibles qui aideront les élèves à s’approprier le langage SQL. Ceci peut aussi servir de base pour une **évaluation**.

Consignes aux élèves : En utilisant la base suivante, écrire les requêtes SQL correspondant aux différents exercices. *Difficulté croissante (les dernières ne sont pas au programme officiellement)*

Voir document joint *etape1.docx* pour une correction.

admin (idAdmin, login, password, nom, prenom)

superieur (idSuperieur, login, password, nom, type, commune, lattitude, longitude, nbAdmis, nbAppel,   
 coefNote1, coefNote2)

lycee (idLycee, login, password, nom, commune)

eleve (idEleve, #idLycee, login, password, nom, prenom, anneeNaissance, note1, note2)

candidature (#idEleve, #idSupérieur, statut)

Exercice1 : Donner tous les noms des lycées

Exercice2 : Donner les noms des écoles du supérieur dans l’ordre alphabétique décroissant

Exercice3 : Donner les 10 premiers noms, prénoms des élèves ayant obtenu au moins 19 en note1

Exercice4 : Donner les prénoms et moyennes des élèves dont le prénom contient la lettre A et dont la moyenne des notes 1 et 2, nommée moyenne, est strictement supérieure à 16

Exercice5 : Donner le nombre d’élèves du lycee ' Lycée polyvalent Aiguerande'

Exercice6 : Donner le nombre d’élèves du lycée ' Lycée général et technologique Jean-Paul Sartre' qui ont postulé pour la faculté de droit

Exercice7 : Donner les noms des filles prénommées 'Charlotte' qui sont dans le même lycée que 'Charlotte Weis' (sauf elle-même !) *auto jointure…ou requêtes imbriquées*

Exercice8 : Donner, pour chaque école du supérieur, son nom et le nombre de postulants *GROUP BY*

Exercice9 : Donner, pour chaque école du supérieur, son nom et le nombre de postulants du lycée du Parc ayant plus de 16 de moyenne *GROUP BY + HAVING*

Exercice10 : Donner, pour chaque école du supérieur, son identifiant et le nombre de postulants du lycée d’identifiant 4 en écrivant 0 si aucun élève n’a postulé *LEFT JOIN + COALESCE*

**Atelier 2 : Requêtes SQL de création de la BD MonAvenir**

**Atelier 3 : Quelles fonctionnalités attendre de notre application ?**

**papier crayon puis DB BROWSER**

Consignes aux élèves :

- Travailler en groupes à partir de la base de données *MonAvenir*

- Chaque groupe d’élèves prend un rôle parmi : administrateur, lycéen, chef d’établissement du supérieur, proviseur d’un établissement du secondaire.

- Selon le rôle endossé, quelles requêtes voudriez-vous pouvoir effectuer ?

- Ecrire ces requêtes en langage SQL.

- Les tester et corriger si besoin en utilisant la BD *MonAvenir* et le logiciel DB BROWSER.

Etape 2 : Utiliser Open Data, Fake pour constituer fichiers sources CSV

**Atelier 1 : Travail sur des données open data et sur le format CSV**

**Génération des données sur les lycées et les établissements du supérieur**

Lors de cet atelier, nous serons amenés à échanger avec les élèves sur :

- les données open-data

- les enjeux éthiques du stockage de données nominatives sur les élèves (d’où des données fictives)

- les normes d’encodage du texte

**1/** **Récupération des données lycées**

Sur internet, saisir l’adresse : [https://data.education.gouv.fr](https://data.education.gouv.fr/), puis cliquer sur l’onglet Données.

Effectuer des filtrages pour obtenir la liste des **154 lycées ouverts du Rhône** (ne pas tenir compte de celui qui va ouvrir ni de celui qui va fermer). 151 lycées ????

Exporter le jeu des 154 données au format CSV sous le nom **lycee-rhone-data.csv**

Ouvrir le fichier avec un tableur et supprimer les colonnes inutiles : garder uniquement **Appellation officielle** et **Commune**.

Il y a davantage de colonne dans notre fichier lycee-rhone.csv

J’ai nom établissement au lieu de appellation officielle

**2/** **Récupération des données supérieurs**

Aller sur le site internet [https://www.data.gouv.fr/en/datasets/etablissements-denseignement-superieur-2/#](https://www.data.gouv.fr/en/datasets/etablissements-denseignement-superieur-2/)

et exporter la liste des établissements d’enseignement supérieur au format csv et renommer sous

**superieur-rhone-data.csv**

Ouvrir le fichier avec un tableur et supprimer les colonnes inutiles : garder uniquement **type d’établissement**, **nom**, **commune, département, latitude** et **longitude**. Avec des tris/filtres, garder uniquement les 181 écoles du Rhône, puis supprimer la colonne département et inverser l’ordre des deux premières colonnes en nommant : **nom**, **type**, **commune, latitude** et **longitude.**

**3/ Modification des données lycées et supérieurs**

Ouvrir et compléter le fichier **generer\_base\_csv.py** dans lequel on utilise le module Faker (qu’il faudra importer si nécessaire) afin de créer des pseudos ou mots de passes fictifs.

Ce fichier contient 3 fonctions : la première generer\_login permet de générer un login en vérifiant que celui-ci n’est pas déjà pris ; les 2 fonctions suivantes permettent de compléter les fichiers CSV pour obtenir les fichiers

**lycee-rhone.csv et superieur-rhone-data.csv**

Compléter la fonction **generer\_login**, puis en vous inspirant de la fonction **generer\_csv\_lycee**, écrire la fonction **generer\_csv\_superieur**.

**Atelier 2 : travail sur python et lecture / écriture de fichiers CSV**

**Génération des données sur les élèves et leurs candidatures**

~~BUT : peuplement des classes eleve, admin et candidature…~~

**1/** **Classe eleve**

- Ecrire une fonction **anneeNaiss(a)** qui simule les années de naissances des élèves de terminales.

Elle prend en argument une année a et renvoie une année entre a et a +4 selon une certaine distribution correspondant aux pourcentage d’élèves ayant des années de retard ou d’avance…

Si a = 2000, la fonction doit renvoyer un nombre entier entre 2000 et 2004 compris, sachant qu’il y a environ 1% d’élèves qui ont 2 ans de retard (nés en 2000), 7% d’élèves qui ont 1 an de retard (nés en 2001), 8% d’élèves qui ont 1 an d’avance (nés en 2003) et 1% d’élèves qui ont 2 ans d’avance (nés en 2004).

- Compléter la fonction **generer\_login** qui permet de générer un login en vérifiant que celui-ci n’est pas déjà pris.

- Compléter la fonction **generer\_csv\_eleve** afin de créer des fichiers csv de 60 élèves pour chacun des lycées.

- Compléter la fonction **generer\_csv\_candidature** ou plutôt generer\_csv\_superieur et generer\_admin ???

Moduler les quotas du nombre d’appel et du nombre d’admis en fonction du type de l’école.

Prévoir un lycée où une école ne sera pas demandée.

Etape 3 : Python et fichiers CSV – Peuplement de la BD *MonAvenir*

Je n’ai pas regardé cette partie

**Atelier 1 : Peuplement à partir d’un fichier .sql (NotePad++)**

Compléter la fonction python **creer\_base\_monavenir** du fichier **peupler\_base\_bd.py**.

Elle prend en argument le fichier bd à créer et le fichier sql à partir duquel on va créer la bd.

**Atelier 2 : Peuplement à partir d’un fichier .csv ici on peut faire 3 groupes !!**

En vous inspirant de la fonction **peupler\_eleve**, compléter les fonctions python **peupler\_lycee** **peupler\_superieur, peupler\_admin,** du fichier **peupler\_base\_bd.py** en insérant les données lues, cette fois, dans les fichiers csv correspondants.

Attention les 3 fonctions sont plus simples et ne nécessitent pas l’utilisation d’un compteur…

**Atelier 3 : Peuplement dynamique à partir d’un script de python**

Compléter la fonction python **peupler\_candidature**

Puis regarder et commenter chaque ligne de la fonction **peupler\_base\_mon\_avenir**

En particulier, ils devront comprendre seul l'insertion d'un élève test...

Etape 4 : Développement WEB côté serveur en Python

Création du formulaire inscription et du fichier main

**Voir le document élève joint** qui détaille l’activité décrite dans les grandes lignes ci-dessous.

**Atelier 1 : découverte de** [**Flask**](https://flask.palletsprojects.com/en/1.1.x/)

**Atelier 2 : formulaire HTML de connexion et traitement par python Flask**

- Réaliser un formulaire HTML de connexion dans la page d’accueil qui fournit un couple (*login*, *password*).

- Programmer une fonction de traitement du formulaire de connexion qui va interroger la base de données pour vérifier l’existence du couple (*login*, *password*) et, selon le succès de la requête et le profil, retourner une page web d’erreur ou une page eleve.html, lycee.html, superieur.html, admin.html.

**Atelier 3 : formulaire de recherche dans la BD et traitement par python Flask**

Une fois la connexion réalisée, on souhaite qu’un utilisateur de profil élève puisse interroger la base : par exemple rechercher les établissements du supérieur par type (il y en a 10 ) et par commune.

Etape 5 : Traitement des candidatures

Non développé. En prolongement. Le code est fourni.

Etape 6 : Création des formulaires de consultations des résultats

**TP noté par groupes** : Le travail préliminaire a donné tous les outils nécessaires aux élèves pour réaliser cette tache finale et donc le travail sera rendu pour **évaluation**.

**Travail sur formulaire consultation HTML et fichier main**

Consigne aux élèves :

- Chaque groupe se voit attribuer un rôle parmi : élève, lycée, établissement du supérieur, administrateur.

- Suivant votre rôle et en vous inspirant de ce qui a été fait en classe pour le formulaire d’inscription et pour le fichier main :

\* créer un formulaire de consultation HTML

\* compléter le fichier main correspondant avec les requêtes SQL sous-jacentes

\* le fichier HTML renvoyé par le serveur en réponse à la requête

On donne une image du formulaire attendu (4 images différentes pour chacun des quatre formulaires)

IMAGES

Etape 7 : CSS

En D.M. ON demande aux élèves de réfléchir à l’UX Design… et de prévoir un CSS pour le projet.

Ce travail peut encore être mené par petit groupe et fera l’objet d’un bonus.