

Projet 5 , Utilisez les données publiques de l'OpenFoodFacts



Dans ce document, nous présenterons :

- ➡ Les fonctionnalités attendues
- ➡ Le choix des algorithmes

La startup Pur Beurre , dispose d'un restaurant et remarque que ses clients souhaitent modifier leur habitude alimentaire, remplacer les produits habituels par des substituts plus sains.

Plusieurs Contraintes et Fonctionnalités nous ont été demandées , parmi les plus importantes nous retrouvons :

- ☐ Utiliser MYSQL et python
- ☐ L'utilisateur doit interagir avec le programme via le terminal
- ☐ Répéter la question si l'utilisateur entre un caractère qui n'est pas un chiffre

Fonctionnalités attendues

Utiliser python et MYSQL

```
# to create a user with creation and modification rights
def create_user(self,):
    try:
        # create 3 variables for create user allowed privilege and actualiz
        user = "CREATE USER 'StudentOF'@'localhost'\
            IDENTIFIED BY '1Ksable$';"
        privilege = "GRANT ALL PRIVILEGES ON * . *\
            to 'StudentOF'@'localhost';"
        flush = "FLUSH PRIVILEGES;"
        self.mycursor.execute(user)
        self.mycursor.execute(privilege)
        self.mycursor.execute(flush)
        print("Création de l'utilisateur")
        # if the user already exists we validate anyway
    except mysql.connector.errors.DatabaseError:
        print("Utilisateur déjà existant")

# to create database
def create_db(self,):
    self.mycursor = self.mydb.cursor()
    data_name = 'CREATE DATABASE IF NOT EXISTS openfoodfact;'
    self.mycursor.execute(data_name)
    print("Création de la base de données")
```

Fonctionnalités attendues

afficher une interface utilisateur

```
1 - Quel aliment souhaitez-vous remplacer ?  
2 - Retrouver mes aliments substitués.  
0 - quitter le programme
```

```
quel aliment recherchez vous ?
```

```
choisir votre catégorie :
```

```
1 - Légumes  
2 - Viandes  
3 - Poissons  
4 - Biscuit et gateaux  
5 - Sandwich  
6 - Fromage  
7 - Chocolats  
8 - Boissons  
9 - Fruits  
0 - Quitter  
>>>
```

Environnement de travail

- ☐ Windows 10
- ☐ Virtualenv
- ☐ Visual Studio Code

Pre-projet

Avant de commencer j'ai du me renseigner sur différentes informations et suivre différents cours :

- Me renseigner sur l'API d'OpenFoodFacts;
- Utiliser le module requests pour pouvoir récupérer les données de l'OpenFoodFacts;
- Réfléchir à la conception de la base de données, qui a été revu plusieurs fois
- Me renseigner sur le fonctionnement des requêtes SQL
- Réfléchir au moyen d'incorporer les données de l'OpenFoodFact dans ma base de donnée



L'algorithme de substitution

Requête se trouvant dans la méthode de substitution

```
def research_sql_substitute(self,
                                cat_modif,
                                id_substitute,
                                name_product,
                                nutriscore_product):
    """we search with a similar category but with a name
    and a different id and we take the best nutrition grade"""
    sql_substitute = "SELECT product_name, Nutrition_grade\
FROM product INNER JOIN categories\
ON product.Categories_id = categories.id\
WHERE categories.Categories LIKE %s\
AND product.id != %s AND product.product_name != %s\
ORDER BY Nutrition_grade ASC LIMIT 1"
    val_substitute = ("%"+cat_modif+"%", id_substitute, name_product)
    self.db.mycursor.execute(sql_substitute, val_substitute)
    return self.db.mycursor.fetchone()
```


L'algorithme de substitution

```
while end_substitute:
    nb = 1
    cat_modif = categ_susbtitute
    print(cat_modif)
    if cat_modif == "":
        print("nous n'avons pas trouvé de meilleur aliment substitué")
        end_substitute = 0
        self.select_choice()
    elif myresult is None or myresult[1] > nutriscore_product:
        list_substitute = cat_modif.split(", ")
        del list_substitute[-nb:]
        categ_susbtitute = ", ".join(list_substitute)
        nb += 1
        myresult = self.research_sql_substitute(cat_modif,
                                                id_substitute,
                                                name_product,
                                                nutriscore_product)
    # if we have one result , ok we can display our aliment
    elif myresult[1] <= nutriscore_product:
        print("voici un aliment substitué : ")
        self.list_aliment.append(myresult[:-1])
        self.choice_alim = 0
        self.display_aliment()
        break
```

CONCLUSION

- Via ce projet , j'ai pu renforcé des compétences en cours d'acquisition et en acquérir de nouvelles :
- Utilisation de plusieurs librairies
- Structuration d'un projet
- Conception d'une base de données;
- Apprentissage requête SQL
- Fonctionnement d'une base de données