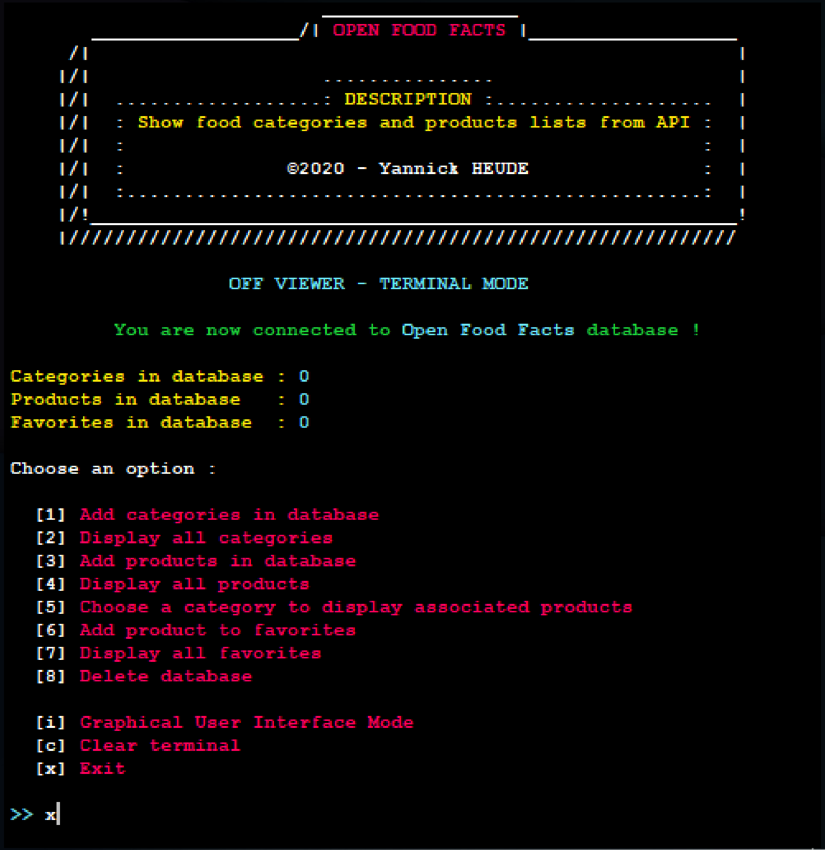
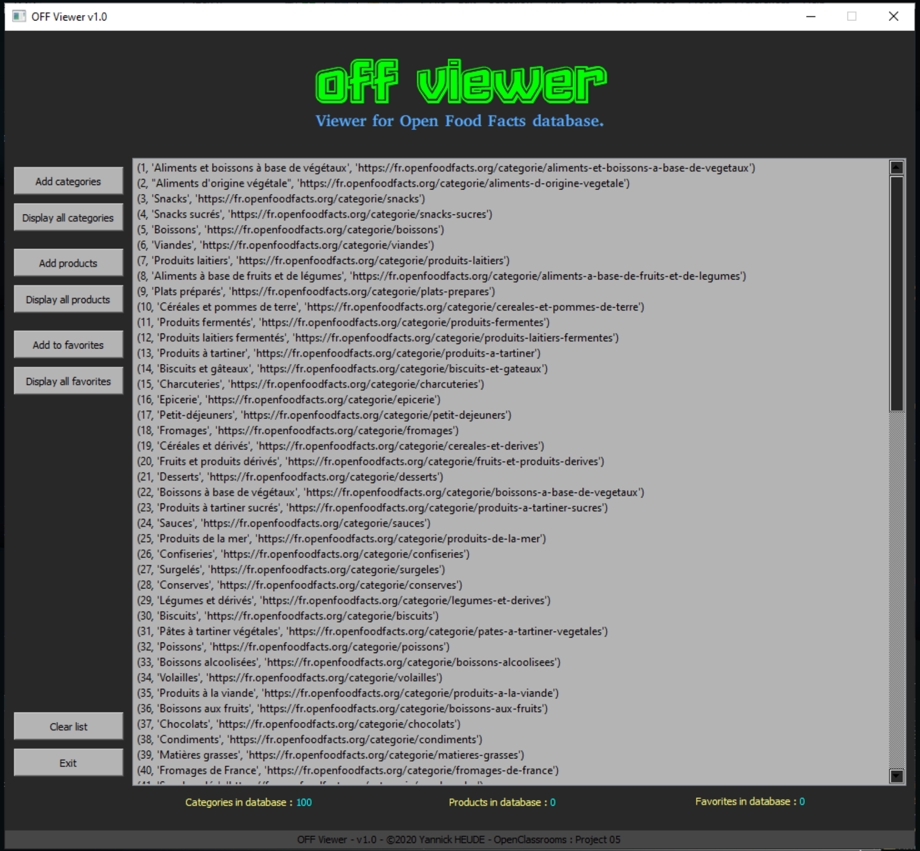
OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts





OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

BUT DU PROJET

- Création d'une base de données
- Récupération des données de l'API Open Food Facts
- Insertion des données dans la base de données
- Gestion et manipulation de la base de données
- Création de 2 programmes (mode Terminal et mode Graphique) pour communiquer avec la base de données

OUTILS ET CONCEPTION

- Base de données :: MySQL
- Programme Graphique :: PyQT5
- Un fichier SQL pour la structure de la base de données
- Un fichier par classe (Category, Product, Favorite)
- Un fichier pour la connexion à la base de données
- Un fichier pour la communication entre la base de données et les programmes
- Un fichier de scripts pour chaque mode
- Un package pour les requêtes vers le site d'Open Food Facts
- Un fichier de gestion des constantes des programmes

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

Le mode Terminal (Interface)

- Manipulation de la base de données via une liste
  - Ajout des catégories
  - Affichage des catégories
  - Ajout des produits
  - Affichage des produits
  - Ajout d'un produit en favori
  - Affichage des favoris
  - Suppression des éléments de la base
- Lancement du mode graphique
- Remise a zéro de la console
- Quitter le programme

Le Code

```python
# ====================================
# =   OFF Viewer - TERMINAL VERSION   =
# ====================================

import os
import sys

from ui_app import UIMode
from tools import constants as cst
from json.decoder import JSONDecodeError
from terminal.scripts import TerminalScript
from terminal.terminal_list import ListOptions


class TerminalMode:

    def __init__(self):
        self.terminal_script = TerminalScript()
        self.terminal_list = ListOptions()

    def main(self):
        self.loop = True
        try:
            try:
                self.terminal_script.display_logo()
                while self.loop:
                    self.terminal_list.print_list()
                    user_choice = input(cst.CYAN + " >> " + cst.WHITE)

                    if user_choice == "1":
                        self.terminal_script.user_choice_one()
                    elif user_choice == "2":
                        self.terminal_script.user_choice_two()
                    elif user_choice == "3":
                        self.terminal_script.user_choice_three()
                    elif user_choice == "4":
                        self.terminal_script.user_choice_four()
                    elif user_choice == "5":
                        self.terminal_script.user_choice_five()
                    elif user_choice == "6":
                        self.terminal_script.user_choice_six()
                    elif user_choice == "7":
                        self.terminal_script.user_choice_seven()
                    elif user_choice == "8":
                        self.terminal_script.delete_database()
                    elif user_choice == "i":
                        UIMode().main()
                    elif user_choice == "c":
                        self.terminal_script.display_logo()
                    elif user_choice == "x":
                        sys.exit(cst.EXIT_MSG)
                    else:
                        print(cst.WRONG_CHOICE_LIST + "\n")

            except KeyboardInterrupt:
                sys.exit("\n\n" + cst.KILLED_MSG)
        except JSONDecodeError:
            print(cst.URL_ERROR)


if __name__ == "__main__":
    os.system('cls')
    TerminalMode().main()
```

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

Le mode Terminal (Scripts)

- Tous les scripts sont rangés dans un seul fichier
- Une liste de 8 options permet à l'utilisateur d'interagir intégralement avec la base de données
- Un fichier Controller permet de la communication entre le programme et la base de données
- Vérification du nombre d'éléments dans la base de données afin de les afficher
- Affichage des éléments
- Mise en favori des produits
- Suppression des tous les éléments

Le Code

```python
# ---------------------------------
# -- TERMINAL SCRIPTS MANAGEMENT --
# ---------------------------------

import time

from terminal.logo import Logo
from controller import Controller
from tools import constants as cst


class TerminalScript:
    """ This class allows to manage all functions for terminal view """

    def __init__(self):
        self.controller = Controller()
        self.logo = Logo()

    def display_logo(self):
        self.logo.logo_connected()
        self.controller.get_number_of_categories_in_db()
        self.controller.get_number_of_products_in_db()
        self.controller.get_number_of_favorite_in_db()

    def loop_for_displaying_items(self, table):
        for line in table:
            print(cst.GREEN + (str(line)), end = " | ")
        print()
        for _ in range(125):
            print(cst.BLUE + ".", end = "")
        print()

    # ===== USER CHOICE MANAGEMENT BLOCK =====
    def user_choice_one(self):
        print(cst.CYAN + "\n You have chosen : Add categories in database")
        print(cst.DB_UPDATE_IN_PROGRESS)
        self.controller.add_categories_in_db()
        print(cst.DB_STATUS + cst.UPDATE_OK)
        time.sleep(3)
        self.display_logo()

    def user_choice_two(self):
        if len(self.controller.get_items_in_category_table('id')) == 0:
            print(cst.EMPTY_CATEGORY_TABLE_MSG)
            time.sleep(3)
            self.display_logo()
        else:
            print()
            for category in self.controller.get_items_in_category_table('*'):
                self.loop_for_displaying_items(category)
            print()

    def user_choice_three(self):
        print(cst.CYAN + "\n You have chosen : Add products in database")
        print(cst.DB_UPDATE_IN_PROGRESS)
        self.controller.add_product_in_db()
        print(cst.DB_STATUS + cst.UPDATE_OK)
        time.sleep(3)
        self.display_logo()
```

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

Le mode Graphique (Interface)

- Structure de l'interface graphique au format XML
- Communique avec un fichier contenant les scripts
- Modifiable avec le programme :: Qt Designer
- Aucun script à l'intérieur de ce fichier pour en préserver l'intégrité

Le Code

```
  </property>
 </widget>
 <widget class="QLabel" name="nb_fav_lbl">
  <property name="geometry">
   <rect>
    <x>874</x>
    <y>837</y>
    <width>50</width>
    <height>20</height>
   </rect>
  </property>
  <property name="styleSheet">
   <string notr="true">color: rgb(0, 255, 255);</string>
  </property>
  <property name="text">
   <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span style=&quot; color:#55ffff;&quot;&gt;0&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
  </property>
 </widget>
 <widget class="QLabel" name="fav_in_db_lbl">
  <property name="geometry">
   <rect>
    <x>760</x>
    <y>840</y>
    <width>111</width>
    <height>16</height>
   </rect>
  </property>
  <property name="text">
   <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span style=&quot; color:#ffff7f;&quot;&gt;Favorites in database : &lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
  </property>
 </widget>
 <widget class="QPushButton" name="add_fav_btn">
  <property name="geometry">
   <rect>
    <x>10</x>
    <y>338</y>
    <width>121</width>
    <height>31</height>
   </rect>
  </property>
  <property name="layoutDirection">
   <enum>Qt::LeftToRight</enum>
  </property>
  <property name="styleSheet">
   <string notr="true">background-color: rgb(180, 180, 180);</string>
  </property>
  <property name="text">
   <string>Add to favorites</string>
  </property>
  <property name="checkable">
   <bool>false</bool>
  </property>
 </widget>
 <zorder>welcome_lbl_2</zorder>
 <zorder>dp_cat_btn</zorder>
 <zorder>welcome_lbl</zorder>
 <zorder>cs_list_btn</zorder>
 <zorder>listWidget</zorder>
 <zorder>exit_btn</zorder>
 <zorder>label_2</zorder>
 <zorder>catg_in_db_lbl</zorder>
 <zorder>nb_catg_lbl</zorder>
 <zorder>dp_prod_btn</zorder>
 <zorder>prod_in_db_lbl</zorder>
 <zorder>nb_prod_lbl</zorder>
 <zorder>add_cat_btn</zorder>
 <zorder>add_prod_btn</zorder>
 <zorder>dp_fav_btn</zorder>
 <zorder>nb_fav_lbl</zorder>
 <zorder>fav_in_db_lbl</zorder>
 <zorder>add_fav_btn</zorder>
</widget>
<resources/>
<connections/>
</ui>
```

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

Le mode Graphique (Scripts)

- L'utilisateur interagit avec la base de données grâce à des boutons
- Un fichier Controller permet la communication entre le programme et la base de données
- Accès aux produits d'une catégorie en double cliquant sur cette catégorie
- Mise en favori du produit grâce à son ID

Le Code

```
# -- BLOCK FOR CATEGORY MANAGEMENT --
def add_categories_in_db(self):
    self.add_items_in_db(self.controller.add_categories_in_db())

def display_category_list(self):
    self.display_lists(self.controller.get_items_in_category_table('id'),
                       cst.EMPTY_CATEGORY_TABLE_UI,
                       self.controller.get_items_in_category_table('*'))

# -- BLOCK FOR PRODUCT MANAGEMENT --
def add_products_in_db(self):
    try:
        self.clear_list()
        self.controller.add_product_in_db()
        self.gui.listWidget.addItem(cst.UI_UPDATE_OK)
        self.display_number_of_items()
    except JSONDecodeError:
        self.gui.listWidget.addItem(cst.URL_ERROR)

def display_product_list(self):
    self.display_lists(self.controller.get_items_in_product_table('id'),
                       cst.EMPTY_PRODUCT_TABLE_UI,
                       self.controller.get_items_in_product_table('*'))

def display_products_from_selected_category(self, item):
    self.category_id = item.text().strip("(").strip(")").split(",")[0]
    self.clear_list()
    for product in self.controller.get_products_by_category_id(self.category_id):
        self.gui.listWidget.addItem(str(product))

# -- BLOCK FOR FAVORITE MANAGEMENT --
def add_product_to_favorite(self):
    item_selected = self.gui.listWidget.currentItem().text().strip("(").strip(")").split(",")[0]
    print(item_selected)
    for fav in self.controller.get_product_by_id(item_selected):
        self.controller.add_product_in_favorite(fav)

def display_favorite_list(self):
    self.display_lists(self.controller.get_items_in_favorite_table('id'),
                       cst.EMPTY_FAVORITE_TABLE_UI,
                       self.controller.get_items_in_favorite_table('*'))

# -- PROGRAM --
def main(self):
    # -- BUTTONS EVENTS --
    self.gui.add_cat_btn.clicked.connect(self.add_categories_in_db)
    self.gui.dp_cat_btn.clicked.connect(self.display_category_list)
    self.gui.add_prod_btn.clicked.connect(self.add_products_in_db)
    self.gui.dp_prod_btn.clicked.connect(self.display_product_list)
    self.gui.add_fav_btn.clicked.connect(self.add_product_to_favorite)
    self.gui.dp_fav_btn.clicked.connect(self.display_favorite_list)

    self.gui.cls_list_btn.clicked.connect(self.clear_list)
    self.gui.exit_btn.clicked.connect(self.gui.close)

    # -- DOUBLE CLICK ON SELECTED ITEM TO DISPLAY ASSOCIATED PRODUCTS --
    self.gui.listWidget.itemDoubleClicked.connect(self.display_products_from_selected_category)

    # -- LABEL EVENTS --
    self.display_number_of_items()

    self.gui.show()
    self.app.exec()
```

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

Le fichier Controller

- Permet la communication entre les programmes et la base de données
- Chaque fonction de ce fichier ne retourne qu'une fonction

Le Code

```
class Controller:

    def __init__(self):
        self.del_db = DeleteDB()
        self.db_product = Product()
        self.db_category = Category()
        self.db_favorite = Favorite()

    def delete_database(self):
        return self.del_db.delete_database()

    # ===== CATEGORY HANDLER =====
    def add_categories_in_db(self):
        return self.db_category.add_categories_in_db()

    def get_number_of_categories_in_db(self):
        return self.db_category.number_of_categories_in_db()

    def get_items_in_category_table(self, key):
        return self.db_category.db_column(key)

    # ===== PRODUCT HANDLER =====
    def add_product_in_db(self):
        return self.db_product.add_products_in_db()

    def get_number_of_products_in_db(self):
        return self.db_product.number_of_products_in_db()

    def get_items_in_product_table(self, key):
        return self.db_product.db_column(key)

    def get_products_by_category_id(self, cat_id):
        return self.db_product.show_products_by_category_id(cat_id)

    # ===== FAVORITE HANDLER =====
    def add_product_in_favorite(self, user_choice):
        return self.db_favorite.add_product_in_favorite(user_choice)

    def get_product_by_id(self, prod_id):
        return self.db_product.show_product_by_id(prod_id)

    def get_items_in_favorite_table(self, key):
        return self.db_favorite.db_column(key)

    def get_number_of_favorite_in_db(self):
        return self.db_favorite.number_of_favorites_in_db()

    # ===== FOR UI MODE =====
    def display_number_of_categories_in_db(self):
        return str(len(self.get_items_in_category_table('id')))

    def display_number_of_products_in_db(self):
        return str(len(self.get_items_in_product_table('id')))

    def display_number_of_favorites_in_db(self):
        return str(len(self.get_items_in_favorite_table('id')))
```

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

Système de connexion à la base de données

- Initialisation de l'état de connexion
- Tentative de connexion avec les identifiants
- Messages de validation ou de refus de connexion
- Vérification de connexion après chaque requêtes vers la base de données pour éviter les tentatives de connexions multiples

Les Codes

```python
class DBConnection:
    """
    Module used for connection to database
    """

    def __init__(self):
        self.is_connected = False
        self.dbconnect = None

    def connect(self):
        try:
            self.dbconnect = mysql.connect(
                host = cst.DB_HOST,
                user = cst.DB_USERNAME,
                passwd = cst.DB_PASSWORD,
                database = cst.DB_NAME)
            self.is_connected = True

        except Exception:
            sys.exit(logo.logo_not_connected())

        return self.dbconnect

    def get_db_connection(self):
        return self.dbconnect
```

```python
class DBProvider:

    def __init__(self):
        self.db = DBConnection()

    def get_db(self):
        if not self.db.is_connected:
            self.db.connect()

        return self.db.dbconnect
```

OPENCLASSROOMS - PROJET 05

Utilisez les données publiques de l'Open Food Facts

MERCI !