PYTHON PROJECT - 1

Project is a console-based "Animal Register Program" written in Python. It allows users to manage a simple database of animals by adding, updating, listing, and deleting entries using an interactive command-line interface.

Project Overview

The program simulates a miniature animal management system. Each animal record stores two pieces of information:

- Scientific name
- Common name

Each record is assigned a unique ID generated automatically when the animal is added.

The data is stored in a Python dictionary named animal_dic, where:

- The key is the animal ID (a randomly generated 4-character string of letters and digits).
- The value is another dictionary containing scientific_name and common_name.

To display the records neatly in a table format, the program uses the PrettyTable library.

Key Features

- Add a new animal:
  Prompts the user for the scientific and common names. A unique ID is automatically generated.
- Delete an animal:
  Allows the user to remove an existing record by entering its ID.
- Update an animal:
  Lets the user modify both the scientific and common names of a record.
- List all animals:
  Displays all stored animal records in a well-formatted table using PrettyTable.
- Exit the program:
  Ends the execution safely using sys.exit().

Code Structure and Function Descriptions

1. print_register()
   Displays all the animals stored in the dictionary in a table format with ID, scientific name, and common name columns.
2. random_id()
   Generates a random 4-character string (letters and digits) used as a unique ID for each new animal.
3. add_animal()
   Takes user input to enter a new animal's information and updates the main dictionary.
4. delete_animal()
   Accepts an animal ID and removes its record from the dictionary after confirmation.
5. update_animal()
   Lets users update the details of an existing record after entering a valid animal ID.
6. exit_program()
   Exits the program with a farewell message.
7. While Loop (Main Menu)
   Continuously presents the user with options (a, d, u, l, e) until they choose to exit, executing the corresponding function for each input.

Example Workflow

1. User runs the program and sees the menu.
2. Chooses A to add a new animal (e.g., "Panthera Leo", "Lion").
3. Chooses L to list all animals.
4. Chooses U to update an animal's name using its ID.
5. Chooses D to delete an animal record.
6. Chooses E to safely exit the application.

```
from prettytable import PrettyTable
import random
import string
import sys

animal_dic = {}

print('\nAnimal register program:'
      '\n1: Enter A or a to add new animal.'
      '\n2: Enter D or d to delete a animal'
      '\n3: Enter U or u to update animal.'
      '\n4: Enter L or l to check list of animals. '
      '\n5: Enter E or e to exit the program.')


# ~~~~~~~~~~~~~~~~~~ Functions(): ~~~~~~~~~~~~~~~~~~~
def print_register():
    x = PrettyTable(["ID", "Scientific Name", "Common Name"])
    for animal_data in animal_dic:
        x.add_row([animal_data, animal_dic[animal_data]["scientific_name"],
                   animal_dic[animal_data]["common_name"]])

    print(x.get_string(title="Animal Register"))


def random_id():
    random_string = ''.join(random.choices(string.ascii_uppercase
                                           + string.digits, k=4))
    return random_string


def add_animal():
    animal_id = random_id()
    scientific_name = input("\nPlease enter the scientific name: ").title()
    common_name = input("\nPlease enter the common name: ").title()
    data = {animal_id: {'scientific_name': scientific_name,
                        'common_name': common_name}}
    if not scientific_name and not common_name:
        print("You must write something!")
    else:
```

```
        animal_dic.update(data)


def delete_animal():
    animal_id = input("\nEnter the animal ID you want delete: ").upper()
    try:
        if animal_id in animal_dic:
            choice = input("Delete (y/n)").lower()
            if choice == "yes" or choice == "y":
                del animal_dic[animal_id]
                print(f"{animal_id} register has been deleted!")
        else:
            print("ID not found. Check list pressing 'L'")
    except Exception:
        print("Something bad happend.")


def update_animal():
    animal_id = input("\nEnter the animal ID you want update: ").upper()
    try:
        # If key in dictionary, if key is equal to ID (animal_id)
        for animal in animal_dic:
            if animal == animal_id:
                choice = input(f"Update register {animal_id}? (y/n):
").lower()
                if choice == "yes" or choice == "y":
                    # Changing names
                    scientific_name = input("Write a new scientific name:
")
                    animal_dic[animal]['scientific_name'] = scientific_name
                    common_name = input("Write a new common name: ")
                    animal_dic[animal]['common_name'] = common_name
                    print("Register updated!")
                    print_register()
                else:
                    print("ID not found. Check list pressing 'L'")
    except Exception:
        print("Something bad happend.")
```

```
def exit_program():
    sys.exit("Goodbye!")


# ~~~~~~~~~~~~~~ User's choise ~~~~~~~~~~~~~~~
while True:
    user_input = input("\nwhat do you want to do? (a, d, u, e, l):
").lower()
    if user_input == "a":
        add_animal()
    elif user_input == "d":
        delete_animal()
    elif user_input == "u":
        update_animal()
    elif user_input == "e":
        exit_program()
    elif user_input == "l":
        print_register()
    elif not user_input:
        print("please, enter something!")
```