Programming in Python

Assignment 3

Patient Data

Part 1)

The program in the file **`patient_data.py`** contains a `Dictionary` named `patients` which contains information about several patients. Each patient is represented as a key-value pair, where the key is a patient ID (e.g., `'P12345'`), and the corresponding value is another dictionary containing the patient's details.

Extend this program to implement the following functions. Upon inspection of the dictionary, you may notice some bad data that will cause the program to raise Python error exceptions, handle these errors appropriately using **Try-Except** in your implementation:

- **`def average_age()`**
  This function returns the average age of patients included in the Dictionary. Hint: You can use `patients.keys()` to get a list of the patient IDs in the dictionary `patients`.

- **`def add_medication(id, med)`**
  This function takes a `String` patient ID and a `String` medication as input and appends the specified medication to that patient's medication list.
  - Use the following code to test this function:

```
meds = ["Aspirin", "Acetaminophen", "Penicillin", "Amoxicillin",
"Ibuprofen", "Hydrochlorothiazide"]
for id, med in zip(patients.keys(),meds):
     print("Medications before: ", patients[id]["medications"])
     add_medication(id, med)
     print("Medications after: ", patients[id]["medications"])
```

- **`def upcoming_appointments(id)`**
  This function takes a `String` patient ID and returns the time difference between the patients last checkup date and their next appointment date.
  - Python provides a datatype for working with Date/Time with the following library:
    `from datetime import datetime`
  - This library can be used to cast a `String` to `Date` in a specified format:

```
date1 = datetime.strptime(patients[id]['last_checkup_date'], '%Y-%m-
%d')
date2 = datetime.strptime(patients[id]['next_appointment_date'],
'%Y-%m-%d')
```

  - And the time difference can be found by simply substracting:
    `time_difference = date2-date1`

(40 Marks)

Part 2)

Imagine you are a systems manager for a hospital. You want to create an interface program for one of your doctors to access information on their patients. To help comply with GDPR you extract the patient's data in `patient_data.py` from the interface program by creating two new modules:

fetch_patient_data.py
In this module, import the dictionary `patients` from `patient_data` and implement the following 'get' functions:

- **`def get_patient(id)`**
  Takes a patient ID and returns the patients `name`, `age` and `gender`.
- **`def get_diagnosis(id)`**
  Takes a patient ID and returns the patients `diagnosis`.
- **`def get_medications(id)`**
  Takes a patient ID and returns the patients `medications`.
- **`def get_allergies(id)`**
  Takes a patient ID and returns the patients `allergies`.
- **`def get_meet_dates(id)`**
  Takes a patient ID and returns the `last_checkup_date` and `next_appointment_date`.
- **`def get_ids()`**
  Returns a list of patient IDs.

(30 Marks)

doctor_interface.py
In this module, import the module `fetch_patient_data` and create an **alias** for it.

Write a program that:

- Prints the list of patient IDs using **`get_ids()`**
- Prints the following list of options:
  `details, diagnosis, allergies, dates`
- Asks the user to enter a patient ID and an option from the previously printed list
- Given the entered option and ID, get the patients details using the appropriate function from `fetch_patient_data`
  - For example, if the user enters `P12345` and `allergies`, the program will print `[Penicillin]`
- If the user enters an invalid patient ID or option, handle this appropriately.

(30 Marks)

Submit a **zip folder** containing your Python files on Moodle to complete the assignment. Cooperation for this assignment is permissible but copying is not tolerated. You must understand your code upon submission.