Core Python
Assessment Test

- **Write a program to demonstrate the student management system.**
- **Prepare demonstration of crud operations with student management system under software development principles**
- **Execution of the code following menu must be displayed.**
- **Make sure each business logic is denoted with appropriate comments and make your code interactive and represent clean and clear output on your console screen**.

```
PS C:\Users\Administrator> & python c:/Users/Administrator/Downloads/Student_management_system.py

           press 1 for Counsellor
           press 2 for Faculty
           press 3 for Student


Enter a role id:
```

- **when user select option 1 then it will display following menu**

```
Enter a role id: 1

            1. Add student
            2. Remove student
            3. View all student
            4. View Specific Student

Enter a choice by counsellor:
```

- **Counsellor can add student , remove student , view all student , view specific student**
- **Accept all values dynamically from user**
- **Store all students information in dictionary format**
- **Make sure this code implements using nested dictionary**
- **make sure specific student can only search by id - any appropriate validation if user entered wrong input - if id doesn't fetch data from dictionary display user does not exist**
- **create separate file for all business logics and make them reusable - use modules concepts for implements above logic**

Counsellor can add details as follows:

```
Enter a role id: 1

            1. Add student
            2. Remove student
            3. View all student
            4. View Specific Student

Enter a choice by counsellor: 1
Enter a Serial Number: 1
Enter a First Name: Priya
Enter a Last Name: Patel
Enter a Contact Number: 7784578568
Enter a Subject: Python
Enter a Marks: 89
enter a fees: 35000
Enter a Subject: Java
Enter a Marks: 85
enter a fees: 40000
```

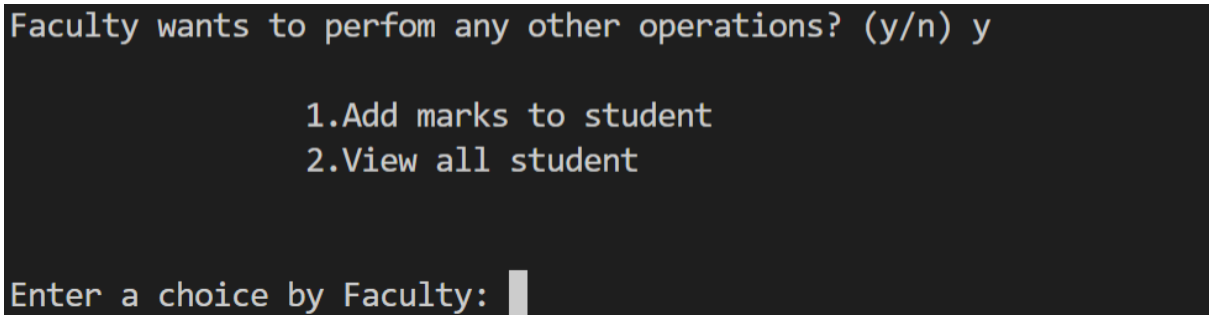
- **Make sure validation proper given - on contact number and first name - display appropriate message if user enter invalid input and accept values again and again - use looping concepts and string inbuilt methods concepts in this logic implementation**

- **Make sure code prevent from unexpected exception**
  E.g. in contact number users can't be able to enter character value if.. Enter Character value - return to the previous menu and accept all details again.

After entered student details output must be like following:

```
{1: {'fname': 'Priya', 'lname': 'Patel', 'contact': '7784578568', 'subject': {
'Python': {'marks': 89, 'fees': 35000}, 'Java': {'marks': 85, 'fees': 40000}},
 'faculty': 'Anjali'}, 2: {'fname': 'Ramesh', 'lname': 'Sharma', 'contact': '8
855446685', 'subject': {'Android': {'marks': 89, 'fees': 74500}, 'Php': {'mark
s': 85, 'fees': 12000}}, 'faculty': 'Nikita'}}
Do you want to perform more operations? (y/n)
```

- **faculty can add students marks - make sure faculty can view and access own students only**
- **Generate a log file and store all transaction details in that log file.**
- **faculty menu as follows :**



- **After each selection menu must be displayed asking for user input**
- **After execution of each option confirmation message must be displayed.**
- **Remove option must ask to user for ID to delete and again ask for confirmation (Y/N) before deletion and display proper message after deletion**
- **Program should not be terminated till the user Exit it**
- **Developer needs to test his product before launching it into the market**
- **After completion this project upload it on GitHub**
    - **Upload all features in develop branch after completion all features merge it with main branch**