Problem Details

The objective of this application is to manage EdYoda Learning Programs using the command line. There will be two types of users:

1. Program Manager: A user type who will have full access to the application. Think of the support team at EdYoda. The support people can perform create, read, update, and delete operations on modules, live sessions, students, etc.
2. Students: A user type who will have permission to only view his/her module(s).

Application Details:

- The program manager should be able to perform CRUD(Create, Read, Update & Delete) operations for modules, units, and students.
- **Modules:**
    a. Modules are technical or non-technical skills. For example, Python, Databases, Data Analysis, etc.
    b. There can be multiple modules in the application.
    c. Module creation: Following details should be required to create a module:
        - ModuleID: Auto-generated unique ID to identify the module.

        - Module Name: It should be a string with a maximum of 30 characters.
        - Module Start Date: Data type should be a date.
        - Module End Date: Data type should be a date.
        - Units: List of all the units in the module.

- **Units:**
    a. These are activities in a module.
    b. A unit can be of any of the following types - live sessions, quizzes, assignments.
    c. *Please note: In this application, the students will not attend live sessions or write quizzes/assignments. You do not have to implement a quiz or assignment feature.*
    d. Unit creation: Following details should be required to create a unit:
        - Unit Id: Auto-generated unique ID to identify the unit.
        - Unit Type: Live Session, Quiz, Assignment.
        - Unit Title: It should be a string with a maximum of 100 characters.
        - Unit Start Date: Data type should be a date.
        - Unit End Date: Data type should be a date.
        - ModuleID: The moduleID of the module this unit will be a part of.

- **Students:**
    a. Students are the users who will be enrolled in one or more modules and can view the list of modules and units.
    b. There can be multiple students in a module. One student can be a part of multiple modules.

Program Manager Features:

- The program manager should see the following options on login:
    a. Manage Modules
    b. Manage Units
    c. Manage Students
    d. Logout
- **Manage Modules:** When the program manager selects this option they should see the following options:
    a. Create a new module
    b. View all modules - This should show the list of all the modules with the following details:
        - ModuleID
        - Module Name
        - Module Start Date
        - Module End Date

        - Units - Return a list of unit IDs that are added to the module
        - Module Status: Upcoming, Ongoing, or Completed based on the module start and end dates. If the module start date is in the future then the status will be Upcoming. If the module end date is in the past then the status will be Completed. If the current date is on or in between the start and end date of the module then the status will be Ongoing.
    c. View module details - This should show all module details based on the ModuleID. The details should include the following:
        - Module Id
        - Module Name
        - Module Start Date
        - Module End Date
        - Units - Return a list of unit IDs that are added to the module
        - Module Status: Upcoming, Ongoing, or Completed based on the module start and end dates.

    d. Update module - Based on the ModuleID, the program manager should be able to update the following details of a module.
        - Module Name
        - Module Start Date
        - Module End Date
    e. Delete module - Based on the ModuleID, the program manager should be able to delete the module along with all the details of it.

- **Manage Units:** When the program manager selects this option they should see the following options:
    a. Create a new unit
    b. View all units - This should show the list of all the units with the following details:
        - UnitID
        - Unit Name
        - Unit Type
    c. View unit details - This should show all details of a unit based on the UnitID. The details should include the following:
        - UnitID
        - Unit Name
        - Unit Type

        - Unit Scheduled Date
        - Unit Start Time
        - Unit End Time
        - Unit Status: Upcoming, Ongoing or Completed based on the unit start and end dates.
    d. Update unit - Based on the UnitID, the program manager should be able to update all the details of a unit. Details that can be edited are:
        - Unit Name(Title)
        - Unit Type
        - Unit Scheduled Date
        - Unit Start Time
        - Unit End Time
    e. Delete unit - Based on the UnitID, the program manager should be able to delete the unit along with its all details.

- **Manage Students:** When the program manager selects this option they should see the following options:
    a. View student details - This should show all the details of a student based on the contact number. The details should include the following:
        - Full Name

        - Mobile Number
        - Email
        - Modules Enrolled - Return a list of module IDs.

For example: Modules Enrolled: [‘2WJW’, ‘HLX9’]

b. Update student - Based on the contact number, the program manager should be able to update all the details of a student.
c. Delete student - Based on the contact number, the program manager should be able to delete all the details of a student.
d. Enroll a Student: The program manager should be able to enroll a student in a module using the contact number.

Student Features:

- Students should see the following options after login:
    a. View Today’s Schedule - If the current date is between the start and end date of a unit then that unit will be visible in the schedule. This unit should be a part of modules that are enrolled by this student. You will have to show the following details:
        - Unit Name
        - Unit Type
        - Scheduled Date

        - Unit Start Time
        - Unit End Time
    b. Update Password
        - Give an option to update his/her password
    c. View My Modules
        - Show a list of all the module IDs the student is enrolled in.
    d. Logout

Common Functionalities:

- Login - Both program managers and students can log in to the application using email and password.
- Registration:
    a. A user can register as a Program Manager or a Student by providing the following details
    b. Following details should be required to register as a student:
        - Full Name: It should be a string with a maximum of 100 characters.
        - Mobile Number: It should be a number with exactly 10 digits. The mobile number needs to be unique for a student which means no two students can have the same mobile number.

        - Email: It should be a string with a maximum of 100 characters. The email address needs to be unique for a student which means no two students can have the same email address.
        - Password: It should be a string with a minimum of 8 characters and a maximum of 16 characters.
- If any of the validations fail then show proper error messages. For example, if multiple validations fail then display “Please enter valid data”. If Mobile Number is not unique then show “Mobile number already exists”
- All the data should be stored in the given JSON files. Store data in students.json if registering as a student and in managers.json file if registering as program manager.

What is already implemented in the base project?

- Registration functionality for both Program Manager and Student is implemented
- Login Functionality for both Program Manager and Student is implemented.
- Manage Modules related functionalities mentioned below are already implemented

    - Create New Module
    - View All Modules
    - View Module Details
    - Update Module
    - Delete Module
- Student Features Functionalities - View Today's Schedule, View My Modules are already implemented.
- Commandline functionality is already implemented. **DO NOT** make changes to this file.
- Helper functions are also provided wherever necessary.
- You can take references from previously implemented functionalities.

What do you have to implement in the project for submission?

- Implement all the remaining functionalities mentioned below
- Manage Units Functionalities: Create New Unit, View All Units, View Unit Details, Update Unit, Delete Unit
- Manage Students Functionalities: Enroll a Student, View Student Details, Update Student, Delete Student
- Please ensure that you update the data in the JSON files. Test cases will verify the JSON files. Even if your functionality is working but you haven't updated the JSON file correctly then the test case will fail and you will not receive any marks.

- All the Program Manager related data will be managed inside the managers.json file. Sample JSON object for the manager:


```
{
  "Full Name": "Test Manager 1",
  "Mobile Number": "1122334455",
  "Email": "testadmin1@gmail.com",
  "Password": "admin1"
}
```


- All the Students related data will be managed inside the students.json file. Sample JSON object for students:


```
{
  "Full Name": "Test Student 1",
  "Mobile Number": "9898989898",
  "Email": "teststudent1@gmail.com",
  "Password": "student1",
  "Modules Enrolled": [
    "N1DF",
    "5N4K"
  ]
}
```


- All the Modules related data will be managed inside the modules.json file. Sample JSON object for module:

```
{
  "Module ID": "5N4K",
  "Module Name": "React",
  "Module Start Date": "2022-02-15",
  "Module End Date": "2022-03-17",
  "Units": [
    "CSB",
    "CL9"
  ],
  "Created By": "Test Manager 2"
}
```


- All the Units related data will be managed inside the units.json file.

  Sample JSON object for unit:


```
{
  "Unit ID": "CSB",
  "Type": "quiz",
  "Title": "DataScience101",
  "Scheduled Date": "2022-01-28",
  "Start Time": "17:00:00",
  "End Time": "19:00:00"
}
```


Instructions for Code Execution:

- **Running the application in Terminal/Command Prompt:**

- Click on Terminal in the top menu
- Click on “Open Terminal in Specific Machine”
- Click on “Applications Machine”
- Goto Projects folder by running the following command “cd Project”
- Open commandline.py file by running the following command “python3 commandline.py”

- **Tips:**
  - When you run the project using the above command in the command line, if you encounter any error then you can read the error, open the commandline.py file see for which function are you getting that error, then open the same function in the operations.py file and fix it.

- **Test Cases:**
  - After you make sure that your code/functionality is working fine and is complete, run the Test Cases from the sidebar.
  - Make sure you don't have any indentation errors or file errors or any type of errors in the code, or else your all test cases will fail.

Important Instructions:

- ***Please ensure that you DO NOT COPY the code from your batch mates or the internet. Our proctoring tool will detect plagiarism and you will receive zero marks in the exam.***
- ***Please DO NOT share your code publicly. If anyone copies your code then both you and the other person will receive zero marks.***