Python Object-Oriented Database Programming Assignment

Dear all:

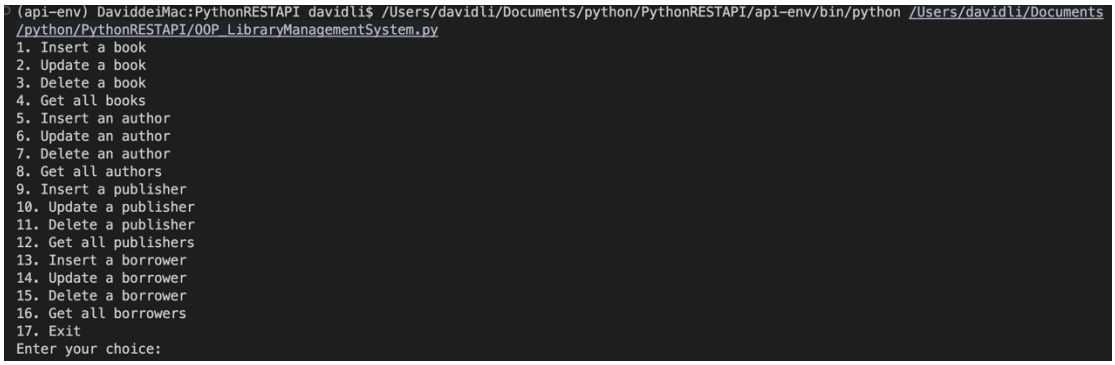
In this assignment, you will create a database management system using Python's object-oriented programming principles. Suppose you're a software engineer and you're assigned a job to create a library database management system. The database will store information about a library system, including books, authors, publishers, and borrowers. You will be required to implement functions for inserting, updating, querying, and deleting data in the database.

1. Basic Part Requirements:

1. Implement a Book class that has attributes like title, author, publisher, publication date, and ISBN number.
2. Implement an Author class that has attributes like name, birth date, and country of origin.
3. Implement a Publisher class that has attributes like name, address, and country.
4. Implement a Borrower class that has attributes like name, email, and contact number.
5. Implement a Library class that will serve as the database management system. This class should have functions for inserting, updating, querying, and deleting data from the database.
6. Implement a CLI interface for interacting with the database management system. This interface should allow the user to perform CRUD (Create, Read, Update, Delete) operations on the library system. That means you should create functions such as
   [1]. Create tables
   [2]. Insert book
   [3]. Update book
   [4]. Delete book
   [5]. Get books
   [6]. Insert Author
   [7]. Update Author
   [8]. Delete Author
   [9]. Get Author
   [10]. Insert Publisher

[11]. Update Publisher
[12]. Delete Publisher
[13]. Get Publishers

The CLI interface could be like as the following interfaces



2. Bonus Part:

Students are encouraged to implement a RESTful API using either Flask or Django Framework that provides the CRUD operations to the library system. The API should allow the user to perform CRUD operations using HTTP requests. In addition, the student's program can deploy the developed application as a web service RESTful API using either Flask or Django Framework. The API should be able to handle multiple requests simultaneously.

If students complete this bonus part, the student's final grades on daily grades (including homework and attendance grades) will be added to 20 points. If the total score on daily work exceeds 100 scores, it will be counted as a 100 score.

3. Submission Requirements:

[1]. Please put your name and student ID at the beginning of your codes for each program

[2]. Please compress your assignment project as a compressed file, a .zip file, and upload the file to the course Moodle website before the due date.

[3]. Late submission is not allowed.

[4]. Please do not copy others' work. If your homework is plagiarized, your grade will be counted as 0 points.

