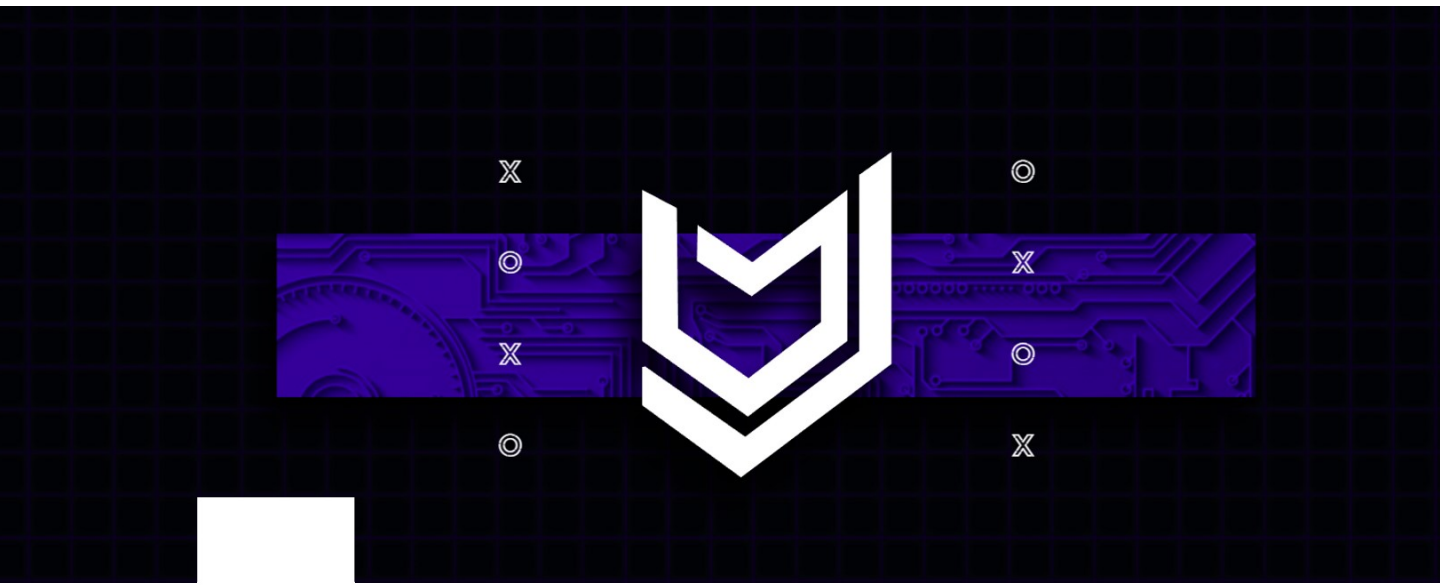

DVM Recruitment Task #1

Library management system

Disclaimer: You are free to design your classes any way you want to. The examples given below are mere suggestions and only one of the many ways to get this task done.

Hello everyone! In this task, you will be building a library management system using some basic principles of Object Oriented Programming.

Class Design

Here is the list of the basic classes, interfaces, and methods one should implement

- Book: A book class that has attributes like name, author, ISBN etc.
  - Add in methods for issuing, returning and reserving books. You will need to use your judgement and decide whether to implement static/class/instance methods or properties.
  - Here's an example code snippet

```
class Book(): def __init__(self): # initialize class # all t
he necessary attributes and details of a book def borrow_boo
k(self): # borrow book def return_book(self): # return book
def reserve_book(self): # reserve book
```

- Shelf: A shelf class for a specific genre that has multiple instances of books stored in an array or dictionary.
  - Add a method to show all the books inside it in a nice formatted way, get counts of all the books on the shelf.
  - Add a method called '**populate_books**' that imports and executes a function imported from another file. More on this later.
  - Here's an example code snippet

```
from .script import populate class Shelf(): def __init__(sel
f): # initialize class # all the necessary attributes and de
tails of a shelf def show_catalog(self): # show a shelf's ca
talogue in a nicely formatted way def add_book(self): # add
a book to the shelf, make sure only Librarian Users have acc
ess to this def remove_book(self): # remove a book from the
shelf, make sure only Librarian Users have access to this de
f get_books_count(self): # additional methods you can think
of def populate_book(self): populate(args) # imported functi
on with necessary args
```

- User: A basic user class that has two roles, and the two roles would have access to different methods.
  - Basic User: A user who can borrow, return and reserve a book.
  - Librarian User: A user who can add/remove a book to/from a shelf and edit a book's details.

populate_books()

- This method will take in a function from a different file and automatically populate a shelf instance with several book instances. The data required for this book instance will be taken from an excel

The data required for this book instance will be taken from an excel sheet.

- You can use a python library like openpyxl to read the spreadsheet data and populate the shelf.
- The excel file will be in the format given below

Name	ISBN	Author
1984	9780451524935	George Orwell

Additional Instructions

- You must use object-oriented principles, and accept and provide values only through the CLI.
- Use python's logger to log each issue and return of a book.
- No property/method should be hardcoded to pass the tests.

Brownie challenge

- Create unit tests that handle every edge and the regular case of use of this program.

Submission

- Deadline: **11th** of **December EOD**.
- Learn some basic git commands and upload your source code on GitHub.
- Submit your task by filling out the form below (use your BITS mail).

Google Forms: Sign-in

Access Google Forms with a personal Google account or Google Workspace account (for business use).

https://forms.gle/hYuzbNRbcA19aV3G8

Resources:

You may refer to the following:

- https://docs.python.org/3/library/unittest.html
- https://docs.python.org/3/howto/logging.html

- https://docs.python.org/3/howto/logging.html
- https://www.geeksforgeeks.org/python-oops-concepts/
- https://www.youtube.com/watch?v=ZDa-Z5JzLYM&list=PL-osiE80TeTsqhIuOqKhwlXsIBIdSeYtc
- https://www.youtube.com/watch?v=HVsySz-h9r4&list=PL-osiE80TeTuRUfjRe54Eea17-YfnOOAx
- https://docs.github.com/en/get-started/quickstart/hello-world
- https://docs.github.com/en/repositories/working-with-files/managing-files/creating-new-files

What to do next?

If you're done with this task, congratulations! You have developed a solid understanding of classes and basic OOP. Now would be a good time to get started with Django!