

Birla Institute of Technology & Science, Pilani Dubai Campus

Principles of Programming Languages

SEMESTER I

2022-2023

**PERSONAL BOOK MANAGER**

SUBMITTED BY:

Chelsy Clevia Fernandes - 2020A7PS0003U

Megha Manoj - 2020A7PS0016U

Mehak Unnisa - 2020A7PS0037U

Shifa Parveen Allaudeen - 2020A7PS0040U

**ACKNOWLEDGEMENT**

We would like to express our special thanks to our faculty Dr. Vijay Kumar who gave us the opportunity to do this assignment and develop our coding skills. This assignment helped us to learn how to use programming language algorithm concepts by using Python in Visual Studio and MongoDB database.

We would also like to thank our supportive colleagues who assisted us in the successful completion of this project.



**TABLE OF CONTENTS**

1. Software & Concepts Used
   1. Visual Studio
   2. MongoDB
   3. Concepts used
2. Problem Statement
3. Algorithm
4. Code
5. Input/output
6. Class Diagram
7. Conclusion
8. References
9. **SOFTWARE & CONCEPTS USED**

The programming language used is Python. The software used are visual studio and mongodb.

**1.1 Visual Studio**

Visual Studio is an IDE that is used to create websites and applications. It runs , debugs the code. It supports many programming languages like C, C++, Python, Ruby and many more.



**1.2 MongoDB**

Mongodb is a document oriented programming database. In our assignment we have used the pymongo library to store and work on the dataset.



* 1. **Concepts Used**

1. Conditional Statements
2. Loops
3. Classes
4. Inheritance
5. File Handling
6. Functions
7. Constructors
8. **PROBLEM STATEMENT**

This Assignment is programmed using python programming language in visual studio code and stores the book in database in MongoDB. The user can add books, update is\_read, delete books, search for a book and display all books.

* It consists of a class named ‘Book’ with attributes bname, author, and genre.
* The class ‘Library’ inherits from the class ‘Book’.
* The class ‘Bookshelf’ inherits from class ‘Library’.
* The execution of the programme starts from main ().
* A loop is created for the user to select the task required to be performed
* To add book to bookshelf
* Update book as read and rate it
* Search a book
* Display all books
* Access Library

1. **ALGORITHM**

A library is created in mongo db that holds the data:

1. library - that contains the data of all the books

2. bookshelf- that holds the details of a user’s personal library

A class called Book stores all the details of all the books.

* It has a constructor that reads the data.
* A print\_details() function that prints all the data.
* A get\_id() function which takes the name of the book and returns the id of the book.

A class called Library that is a subclass of Book that manages all the books in the library.

   Functions:

* set\_rec() - takes the book details, sets the borrow date and return date, adds to database.
* get\_stat() - takes the id of a book, finds the book from database and returns its availability status.
* enter\_library\_data() - used to enter the details of the book into the database.
* update\_status() - to update the availability status of the book, which signifies if it is borrowed or not.
* print\_all() - prints all the books and its details present in the database.
* get\_record() - takes in the name of the book and returns the details of the book.
* del\_rec() - to delete records from library database.
* get\_date() - takes the book name, finds it in the database, and returns the return date of the book.

A class called Bookshelf that is a subclass of Library. This is the personal library of the user. It stores the books the user has borrowed from the library along with its return date, read status, and rating given by the user. The data is stored in an .csv file.

Functions:

* add\_book() - takes in the username and book name, adds from the Library to the Bookshelf by storing it corresponding to the username.
* insert\_into\_file() - takes the book details in the form of array and writes it into the given file.
* update\_rec() - to update the status of the book~~.~~ and hence rate it.
* print\_rec() - retrieves all books from the bookshelf database and
* calls the insert\_into\_file() function to write it onto the given file.

Function called menu()- to perform various functions in the library:

              1. add records

              2. update status

              3. delete a record

              4. search a book

              5. display all records

Function called main() that is used by the user to perform various functions.

1. add book to bookshelf
2. update book if read
3. display the userlist
4. search a book
5. Display all book in the users bookshelf
6. Access the library
7. **CODE**

from pymongo import MongoClient

import pandas as pd

import os

import datetime as dt

client= MongoClient("localhost",27017)

mydb = client["POPL"] #database

collection = mydb["library"] #collection

bookshelf = mydb['bookshelf']

import csv

class Book:

    def \_\_init\_\_(self, bname, author, genre):

        self.name= bname

        self.author= author

        self.genre= genre

    def print\_details():

        print("name: ", self.name,"\nauthor: ", self.author)

        print("genre: ")

        s= " "

        for x in self.genre:

            s=s + " " + x

        print(s)

    def get\_id(self, name):

        q = {'name' : name.lower()}

        doc=  collection.find(q)

        id = doc[0]['\_id']

        return id

class Library(Book):

    #status = False

    def \_\_init\_\_(self, \*args ):

        if len(args)>0:

                self.set\_rec(args)

        else: #empty constuctor

                self.lib()

    def lib(self):

        pass

    def set\_rec(self, args):

        self.bname= args[0]

        self.author= args[1]

        self.genre = args[2]

        x = dt.date.today()

        self.borrow\_date = str(x)

        y = x + dt.timedelta(days=30)

        self.ret\_date = str(y)

        super().\_\_init\_\_(self.bname , self.author , self.genre)

        self.status = False

        self.enter\_library\_data()

    def get\_stat(self, id):

        q = { "\_id": id }

        doc=  collection.find(q)

        stat = doc[0]['status']

        return stat

    def update\_status(self, bname):

        i = self.get\_id(bname)

        q = { "\_id": i }

        stat= self.get\_stat(i)

        if stat == False:

            self.status= True

            new = { "$set": { "status": self.status } }

        else:

            self.status= False

            new = { "$set": { "status": self.status } }

        collection.update\_one(q, new)

        print("taken status updated from ", stat, " to ", self.status)

 def enter\_library\_data(self): #create collections(records) for each

 book with resp. details in mongodb database

        doc1= {

            'name' : self.bname.lower(),

            'author': self.author.lower(),

            'genre' : [x.lower() for x in self.genre],

            'status' : self.status,

            'borrow date' : self.borrow\_date,

            'return date' : self.ret\_date

             }

        collection.insert\_one(doc1)

    def print\_all(self):

        col = collection.find({})

        for x in col:

            for i,j in x.items():

                print(i," : ", j)

            print()

    def get\_record(self, name):

        q = { "name": name }

        doc=  collection.find(q)

        return doc

    def del\_rec(self, name):

 q = { "name": name.lower() }

        collection.delete\_one(q)

         print("deleted")

   def get\_dates(self , b):

     q = { "name": b.lower() }

     doc=  collection.find(q)

     r\_date = doc[0]['return date']

     return r\_date

class BookShelf(Library):

      def \_\_init\_\_(self):

      pass

def add\_book(self, user, bname):

      rec = self.get\_record(bname)

      if rec is None: print('record not found')

      else:

                d = {

                    'user' : user,

                    'bname': rec[0]['name'],

                    'author': rec[0]['author'],

                    'ret\_date': rec[0]['return date'],

                    'is read' : False,

                    'rating': 0

                }

                if (rec[0]['status'] == False):

                     self.update\_status(bname)

                bookshelf.insert\_one(d)

                self.insert\_into\_file(d)

      def insert\_into\_file(self, d):

      f0 = open('bookshelf.csv', 'a')

      x = [d['user'], d['bname'], d['author'], d['ret\_date'], d['is read'],

 d['rating']]

      obj = csv.writer(f0, lineterminator= '\n')

      obj.writerow(x)

f0.close()

def update\_rec(self, user, bname):

      q = {'user' : user,

                'bname': bname.lower()

                }

       curr = bookshelf.find\_one(q)

        if curr['is read'] == False:

       inp = input('mark as read(y/n): ')

if inp == 'y':

     rate = float(input('give rating: '))

     new = { "$set": { "is read": True }, "$set": { "rating": rate } }

     bookshelf.update\_one(q, new)

     else:

     print('not read')

     print("updated bookshelf")

     self.print\_rec()

     def print\_rec(self):

     col = bookshelf.find({})

     for x in col:

     self.insert\_into\_file(x)

     def print\_file(self):

     f1 = open('bookshelf.csv', 'r')

     rec = csv.reader(f1)

     for r in rec:

     print(r)

     #print(('%-10s %-10s %-10s %-10s %-10s %-10s')%(r[0],r[1],

r[2], r[3], r[4], r[5]))

     f1.close()

     def menu(self):

         c = True

     while(c):

       print("Menu")

      print(" 1. add records")

      print(" 2. update status")

      print(" 3. delete record from book")

      print(" 5. Search a book")

      print(" 6. Display all books")

       x = Library()

       choice = int(input("enter option: "))

        if choice == 1:

                print(" ADD RECORD ")

                name = input('Enter book name: ')

                author= input('Enter author name: ')

                genre= input('Enter genres: ')

                genre = genre.split(',')

                x1 = Library(name, author, genre)

            elif choice == 2:

                print("UPDATE BOOK STATUS ")

                i=input("Enter book name :")

                #bid = x.get\_id(i)

                x.update\_status(i)

            elif choice == 3:

                print("DELETE BOOK ")

                name = input("Enter name of book to delete: ")

                x.del\_rec(name)

            elif choice == 5 :

                print("SEARCH ")

                name= input('name of book: ')

                arr = x.get\_record(name)

                for k in arr :

                    for i,j in k.items():

                                print(i," : ", j)

            elif choice == 6:

                x.print\_all()

            else:

                print(" INVALID CHOICE ")

  yn = input('Continue? (y/n)')

if yn == 'y': c= True

else:

 c = False

def main():

  x =  BookShelf()

        usern = input('enter username: ')

        c = True

        while(c):

                print("Menu")

                print(" 1. add book to bookshelf")

                print(" 2. update book as read ")

                print(" 3. display userlist")

                print(" 4. Search a book")

                print(" 5. Display all books")

                print(" 6.  Access Library")

                ch = int(input('enter choice:  '))

                if ch== 1:

                    print('available books:' )

                    x.print\_all()

                    bn = input('enter bookname:' )

                    x.add\_book(usern.lower() , bn.lower())

                elif ch ==2:

                    bn = input('enter bookname:' )

                    x.update\_rec(usern.lower() , bn.lower())

                elif ch==3:

                    x.print\_file()

                elif ch==4:

                    bn = input('enter bookname:' )

                    arr = x.get\_record(bn)

                    for k in arr:

                        for i,j in k.items():

                                print(i," : ", j)

                    print()

                elif ch ==5:

                    x.print\_all()

                elif ch == 6:

                    x.menu()

                else:

                    print(" INVALID CHOICE ")

                yn = input('Continue? (y/n)')

                if yn == 'y': c= True

                else:

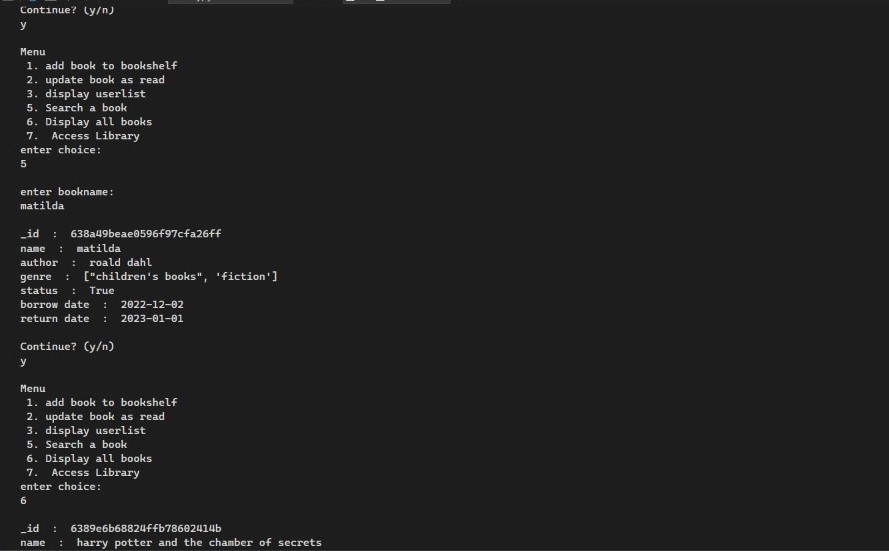
                    print('exiting main menu')

                    c = False

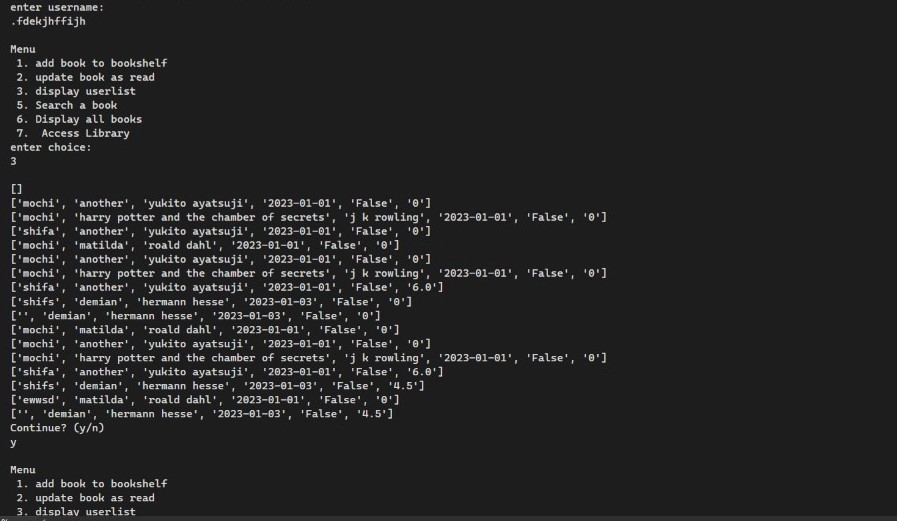
main()

1. **INPUT/OUTPUT**

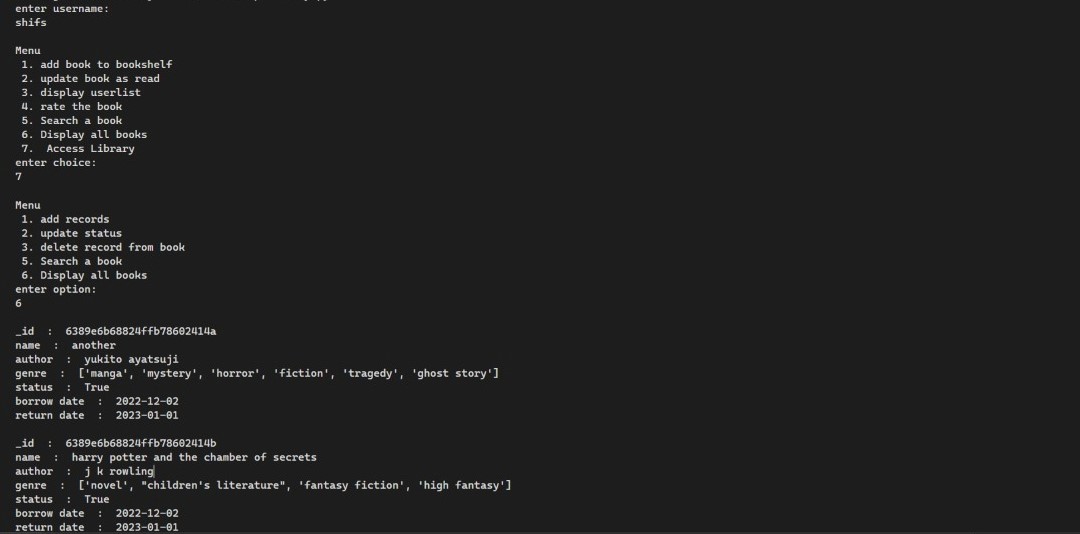
* Enter choice as 5 which will search for a book based on the book name from the bookshelf.

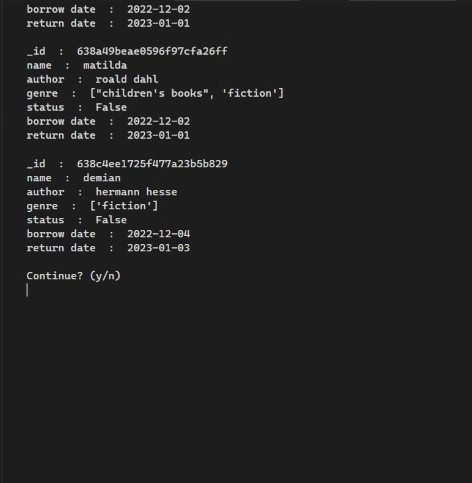
****

* Enter a username and then enter choice as 3 which will display all the user list with the corresponding book name, borrow date ,rate

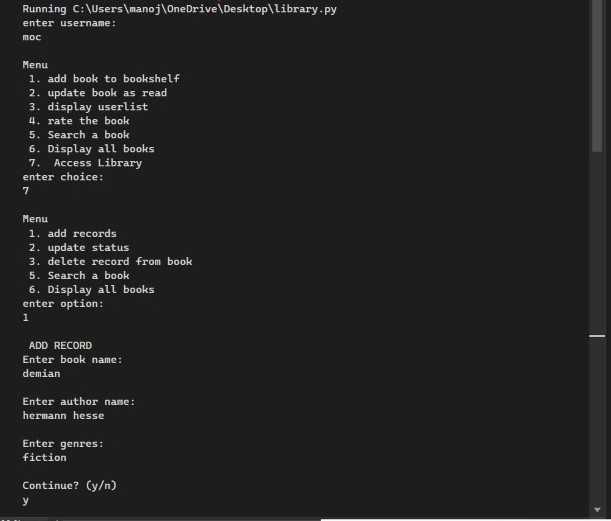
****

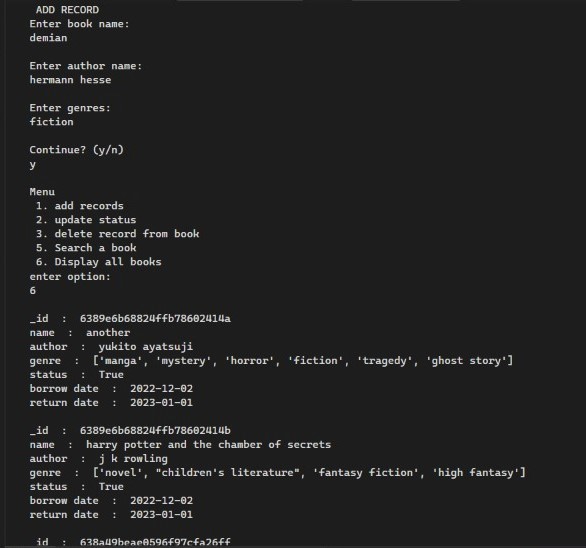
* Enter the username and enter choice as 7 which will let the user to modify the records and select 6 to display all the books available in the record.

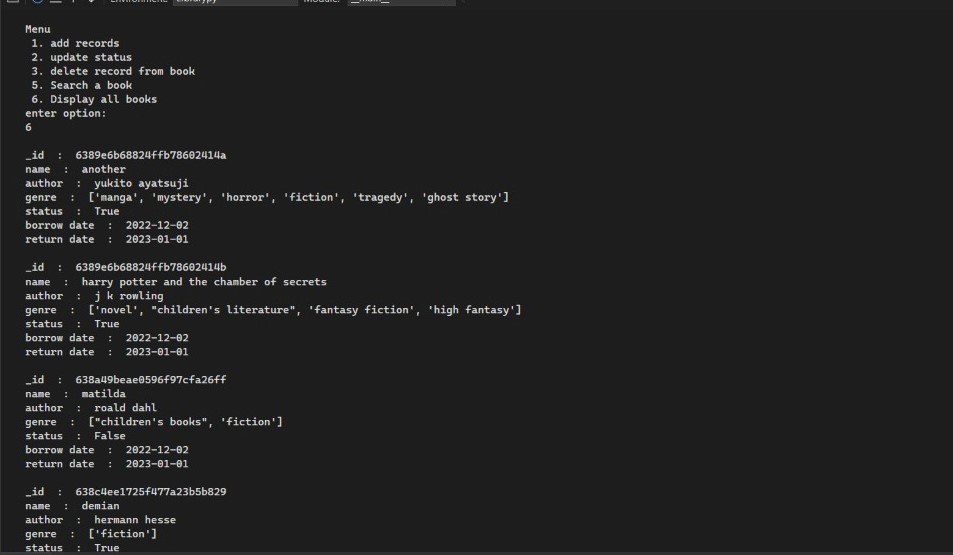
****

****

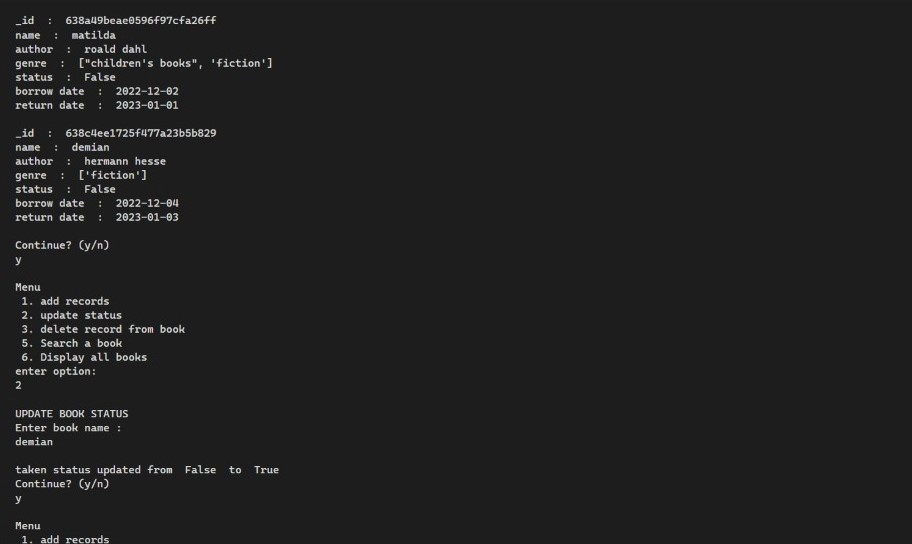
* Enter the choice as 1 to add a book to record and enter 6 next to display all the books added which displays the book added by the user in choice 1.

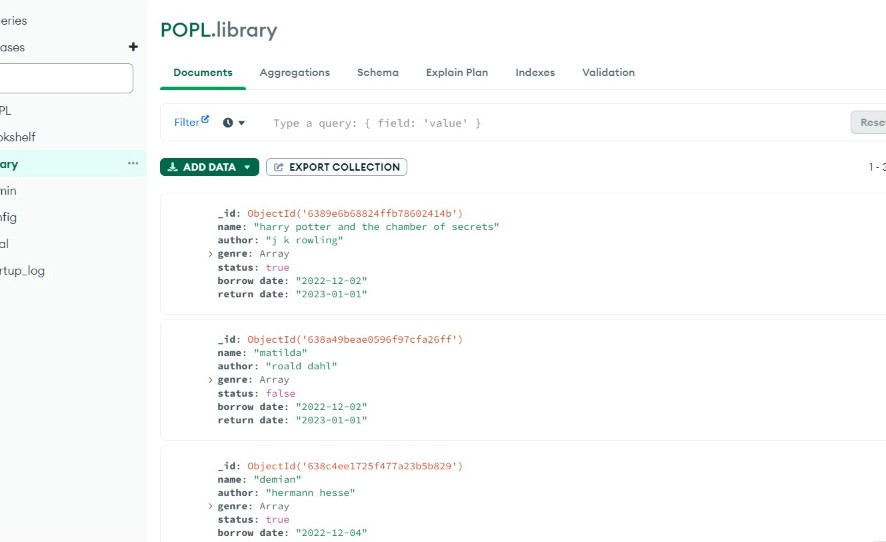
****

****

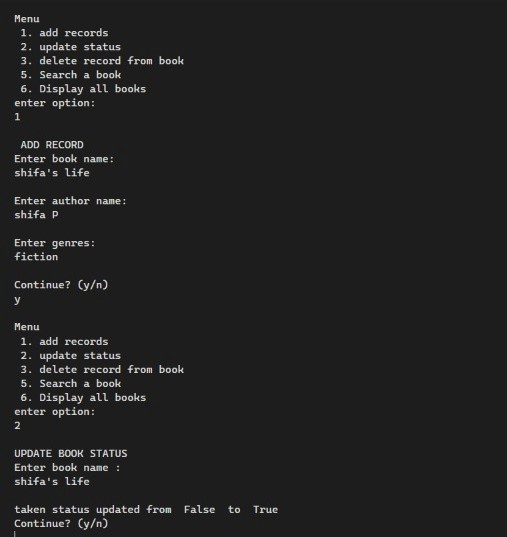
****

* Enter the option as 2 to update the status of a book in the database by inputting a book name.

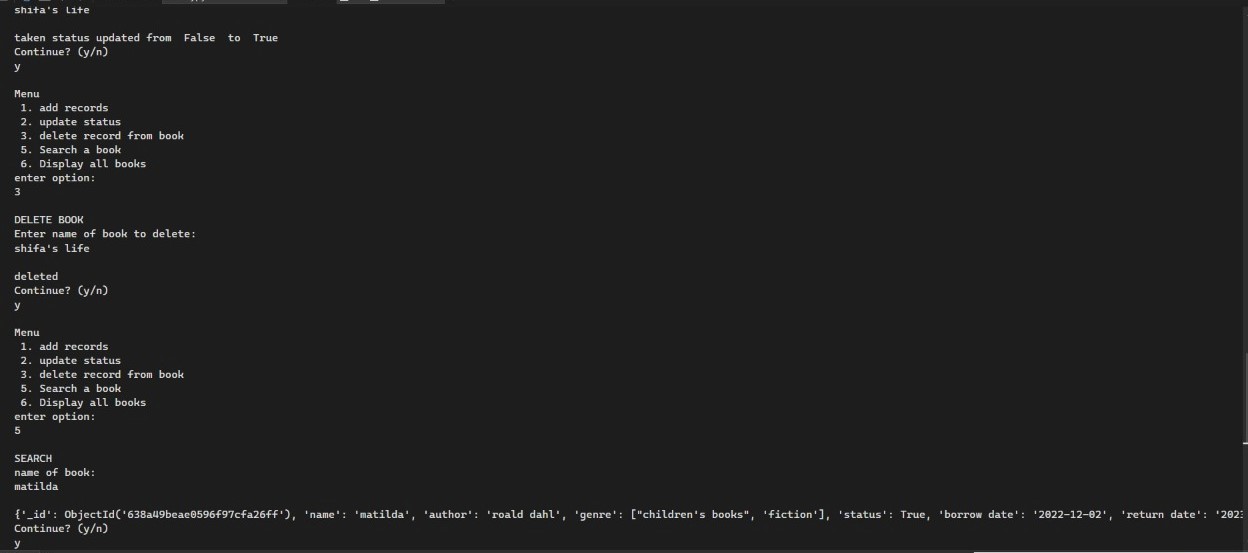
****

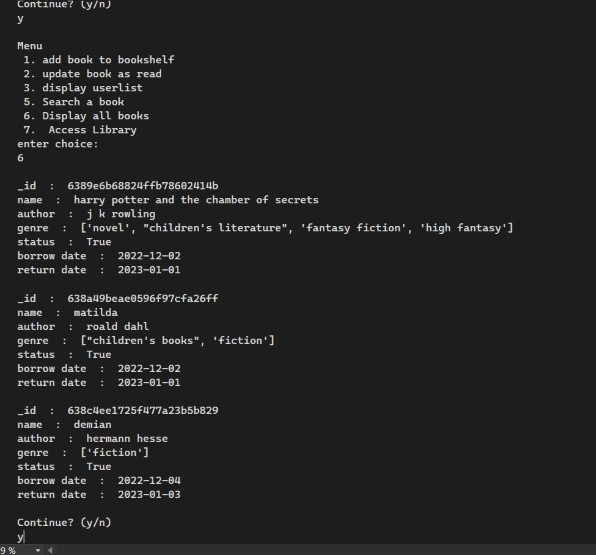
****

* Enter the option as 1 to add another record and then continue the code by inputting ‘y’ and enter option 2 which will update the status of the book.

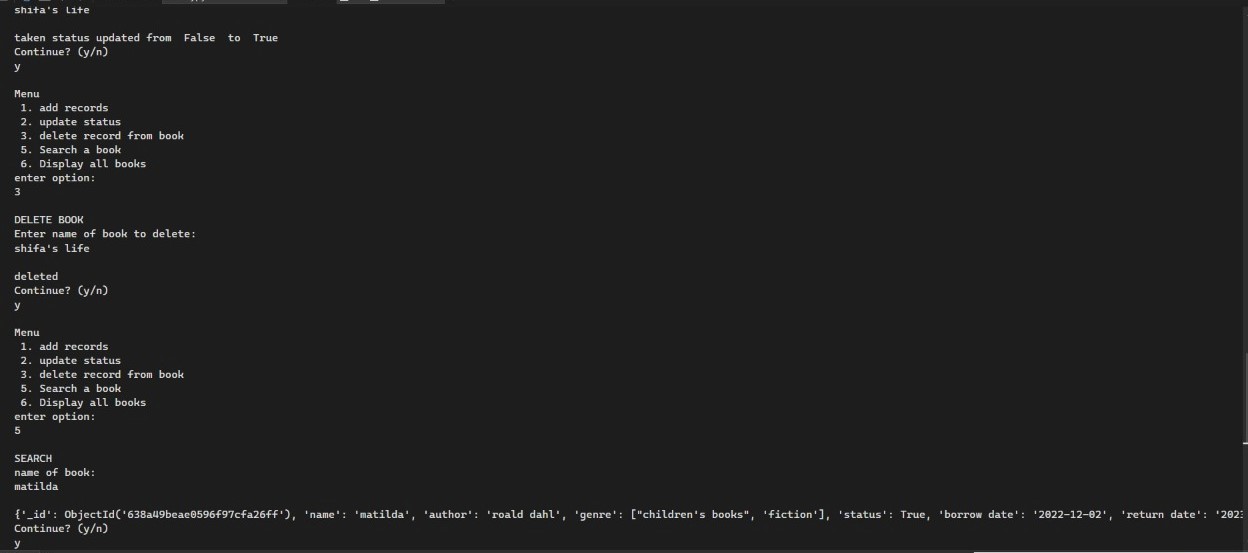


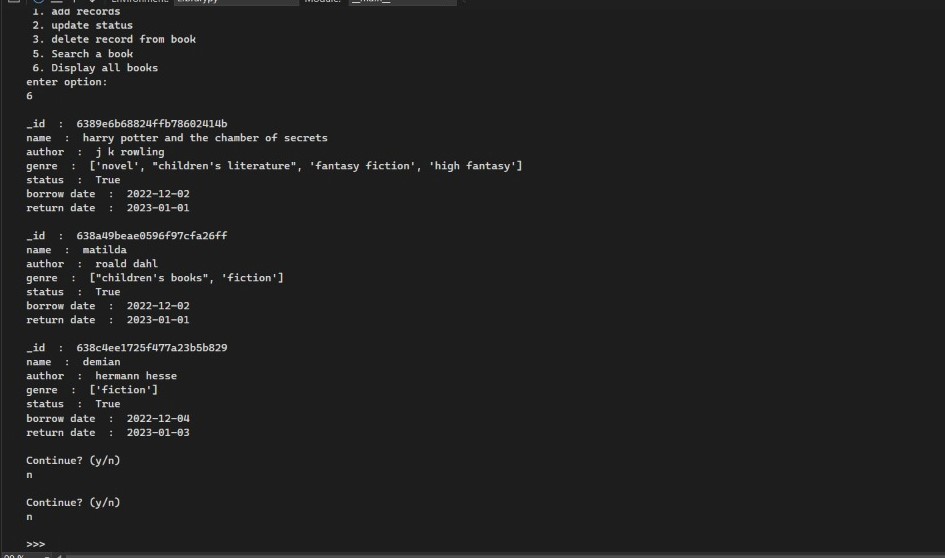
* Enter the option as 3 which will allow the user to delete a book from the record and then continue the code by inputting ‘y’ and enter option 6 to view the booklist and notice the book details of the name inserted in option 3 to be deleted .

****

****

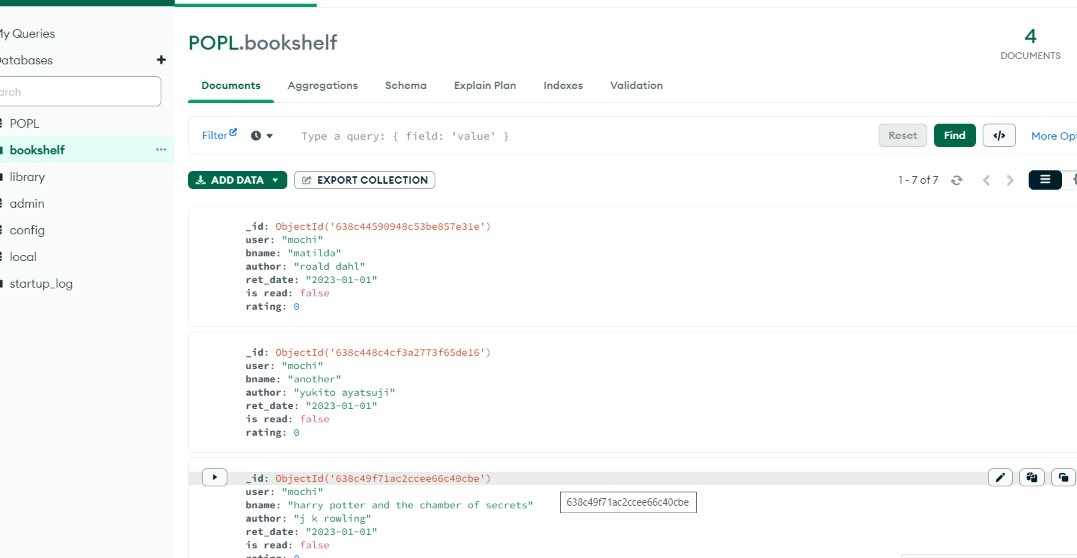
* Enter the option as 5 to search for a book by entering the book name, the output is displayed which includes the details of the book .

****

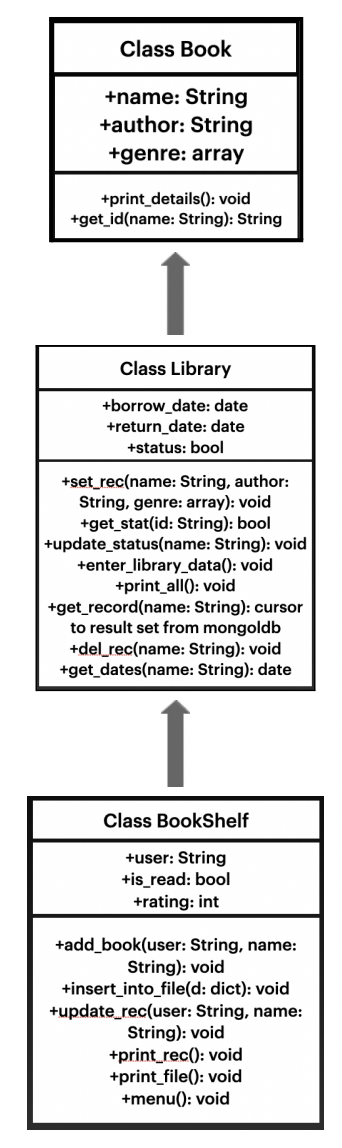
* Exit the main menu by inputting ‘n’ in the continue statement.****
* Books written to a csv file.

****

* Bookshelf database in MongoDB

****

1. **CLASS DIAGRAM**

****

1. **CONCLUSION**

This Assignment was a great opportunity for us to learn to programme

different concepts using python language in visual studio code and access database from MongoDB. It gave us a detailed insight on

the capabilities of the concepts involved in python software and we are

grateful for that. The usage of such python related programs to perform actions, such as a ‘Personal Book Manager’ as done in this assignment, makes the application of python in real life situations simple. The combination of various programs helped in achieving the desired output. We were able to

compile the code using various programming language concepts successfully.

1. **REFERENCES**

* <https://www.mongodb.com/developer/code-examples/python/python-crud-mongodb/>
* <https://www.digitalocean.com/community/tutorials/how-to-perform-crud-operations-in-mongodb-using-pymongo-on-ubuntu-20-04>
* <https://www.google.com/amp/s/www.freecodecamp.org/news/file-handling-in-python/amp/>
* <https://www.programiz.com/python-programming/file-operation>
* <https://code.visualstudio.com/docs/languages/python>
* https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwjUztmtz-D7AhU1RpEFHaldBHYYABAAGgJscg&ohost=www.google.com&cid=CAESauD2tn4FshQfS0qGmFlUM-P4vpyJNn9oXWzUWApNWl2uCCZN2qbpRwLgDLD7-lmbuqMLEXn4N1eTpWoV8JQJEAkctdo7HQM5slZfGaqsdfD1my9u-gUaNxf6KWZaKKkbiWF57vNhyVnP0V0&sig=AOD64\_2sGNum9ZOmXTGqxh4fixA\_Xz3tKg&q&adurl&ved=2ahUKEwiE9NKtz-D7AhW4TqQEHUZ4Cq4Q0Qx6BAgHEAE