Library

Management System

(Computer Project)

Made By

Swarnima ( )

Divya Choudhary ( )

Session : 2020-21

Submitted to

Mr. Rakesh Kumar

D.A.V. Centenary Public School, Chander Nagar, GZB

Phone: 0120-2641049, Email: davcpscn@gmail.com

Certificate

This is to certify that the “Library management system” is made by ………Swarnima…………....Roll No ………………….and …………….Divya Chaudhary…………………….. Roll No …………………………………….…. Under my supervision and Guidance. This is their authentic work for the partial fulfillment of computer project work under C.B.S.E. computer Practical Examination for the session 2020-21.

(Rakesh Kumar)

Acknowledgement

This is the right time to express our gratitude towards our teachers/ mentor for their constant support and guidance. These are only few words which we have written here to show our respect to our teachers.

Though they will not among us after this session but their lesson will always guide us in future also.

This project is our , but it is their inspiration and constant push that we are now able to compile the whole knowledge we learnt through out 11 & 12, in the form of this project.

Last but not least sir we would love to say thank you.

Yours Sincerely

[Swarnima   & Divya]

Index

About Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace.

A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power.

On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

About Library Management System

Library management is a simple computer program designed to help school library to manage books and members records.

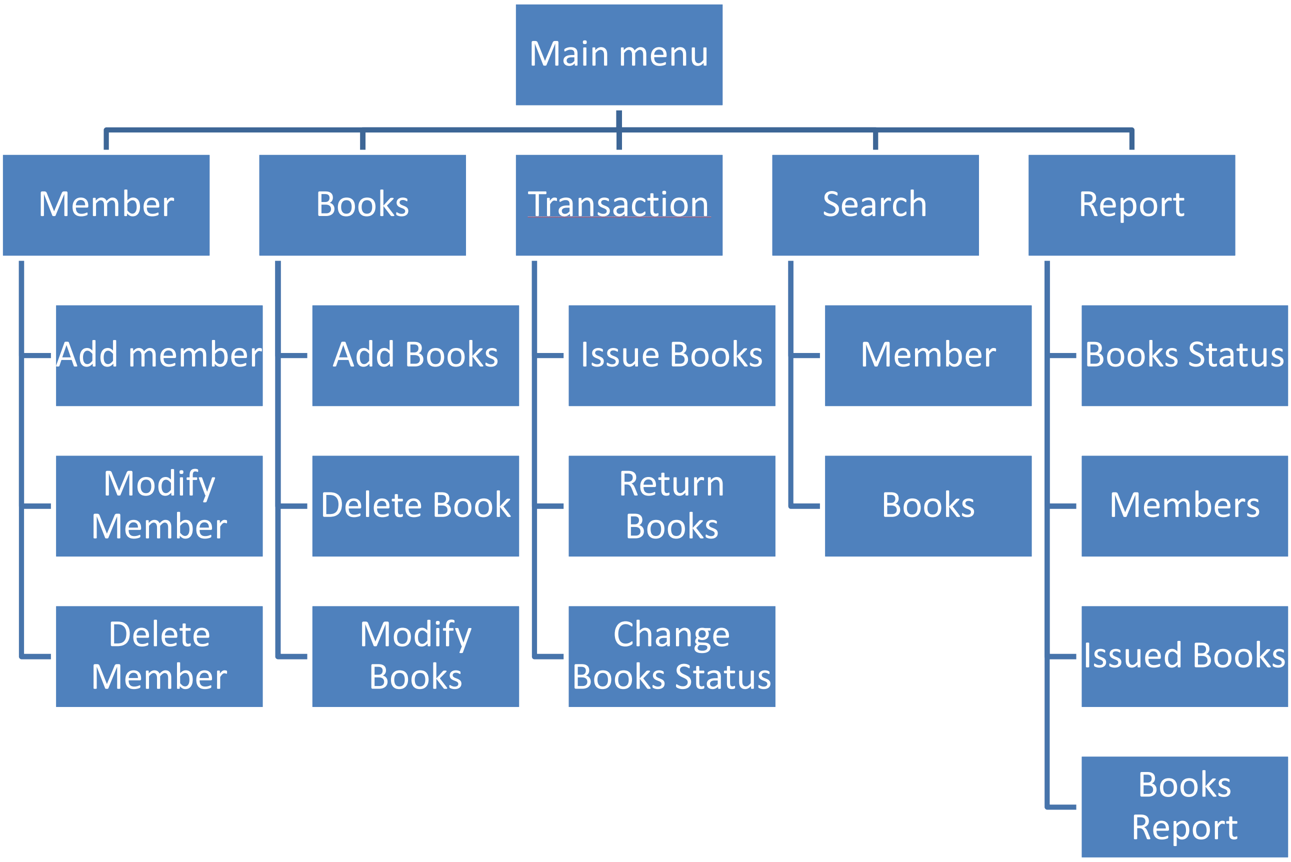
The whole program is designed in this way that you can expand it up to any level. Initially it is used to feed students information and book information along with the transaction status.

This information is recorded in a mysql database. so that this recorded information can be used later on at any stage.

Modification , deletion and searching , reporting facility is given in this project for easy management.

This program is fast and accurate, since it is written in one of the fastest growing programming language ie Python.

Hierarchical diagram of Project



Data Required as Input

Book information

Id	- Unique book Accession Number
Book Title	- Book title
Book Author	- Name of the author(s) of the book
Pages	- Total number of pages in this book
Book Price	- price of the book in INR
Book status	- current book status ie – available, issued, lost,weed out
Book Publisher	- Publisher of the book
Edition	- Edition number of the book

Student Information

Id	- unique Member id
Name	- full name of the member
Class	- Class and section of the member
Address	- Complete address of the member
Phone	- Phone number or mobile number
Email	- any valid email id

Output Required

- Books information
- Member’s information
- Issue books information
- Book’s status report
    - Weed out
    - Lost
    - Available
    - Book in circulation
- Issue Book
- Return Book
- Search
    - Search –Book
    - Search – member
    - Search -Email

Hardware & software requirement

Hardware required

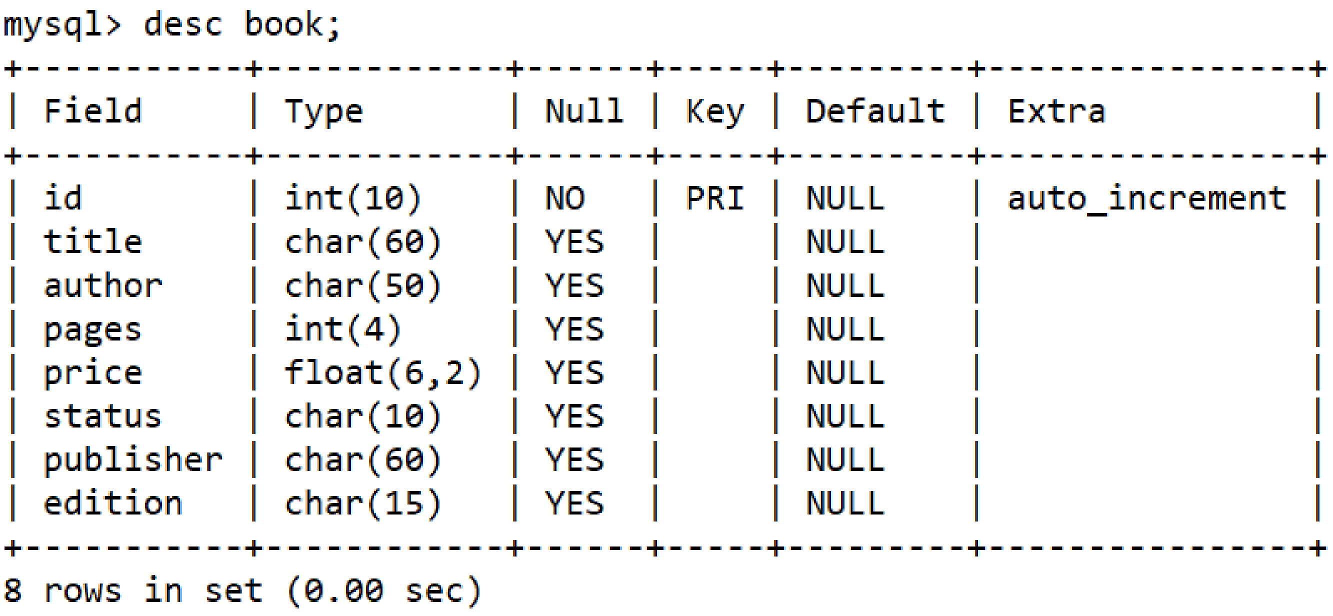
- 4 GB RAM
- Intel i3 or equivalent processor or Above Processor
- Mouse
- Keyboard

Software Requirement

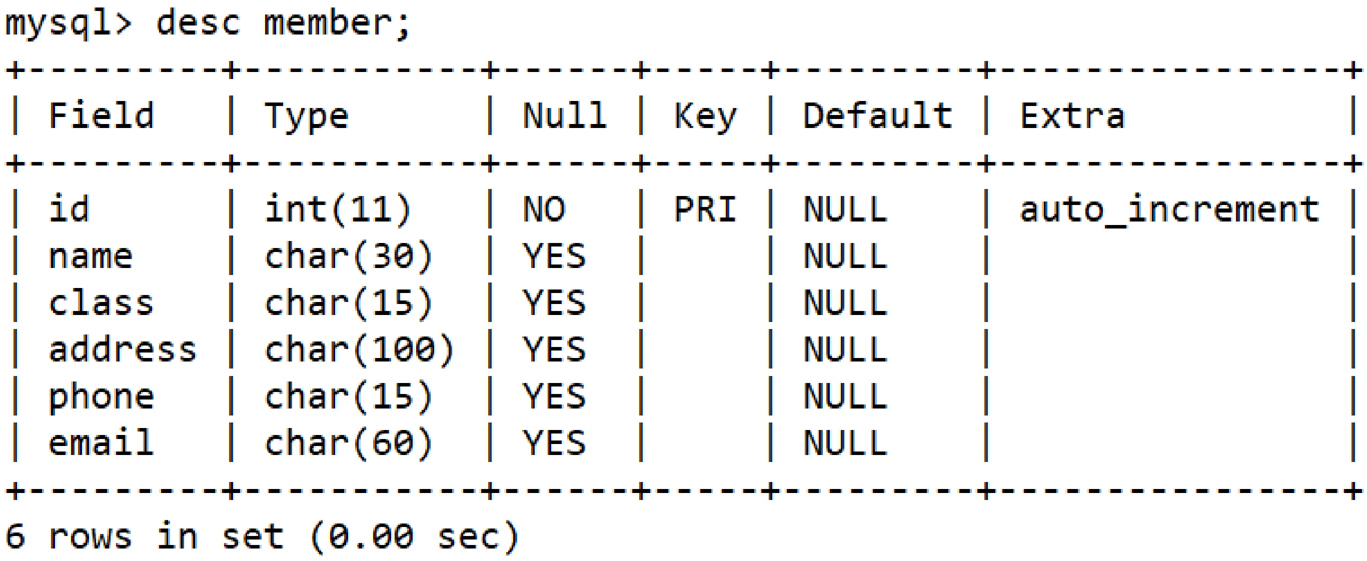
- Operating System – Windows 7 and above or any other supported operating system
- Python Interpreter
- Text Editor (Integrated Development Environment) – Visual Studio Code or similar Text Editor

MySQL
Data Tables

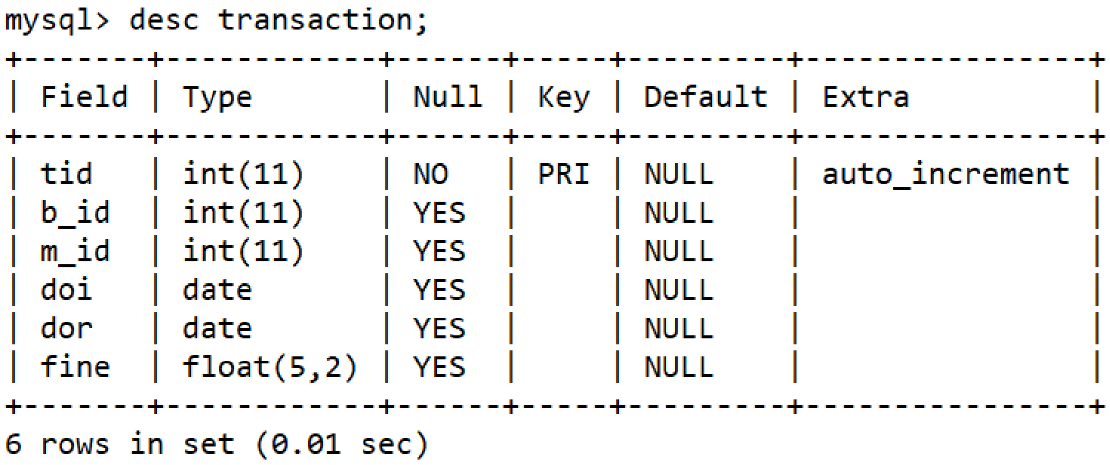
Table Structures



```
mysql> desc book;
+-----------+------------+------+-----+---------+----------------+
| Field     | Type       | Null | Key | Default | Extra          |
+-----------+------------+------+-----+---------+----------------+
| id        | int(10)    | NO   | PRI | NULL    | auto_increment |
| title     | char(60)   | YES  |     | NULL    |                |
| author    | char(50)   | YES  |     | NULL    |                |
| pages     | int(4)     | YES  |     | NULL    |                |
| price     | float(6,2) | YES  |     | NULL    |                |
| status    | char(10)   | YES  |     | NULL    |                |
| publisher | char(60)   | YES  |     | NULL    |                |
| edition   | char(15)   | YES  |     | NULL    |                |
+-----------+------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

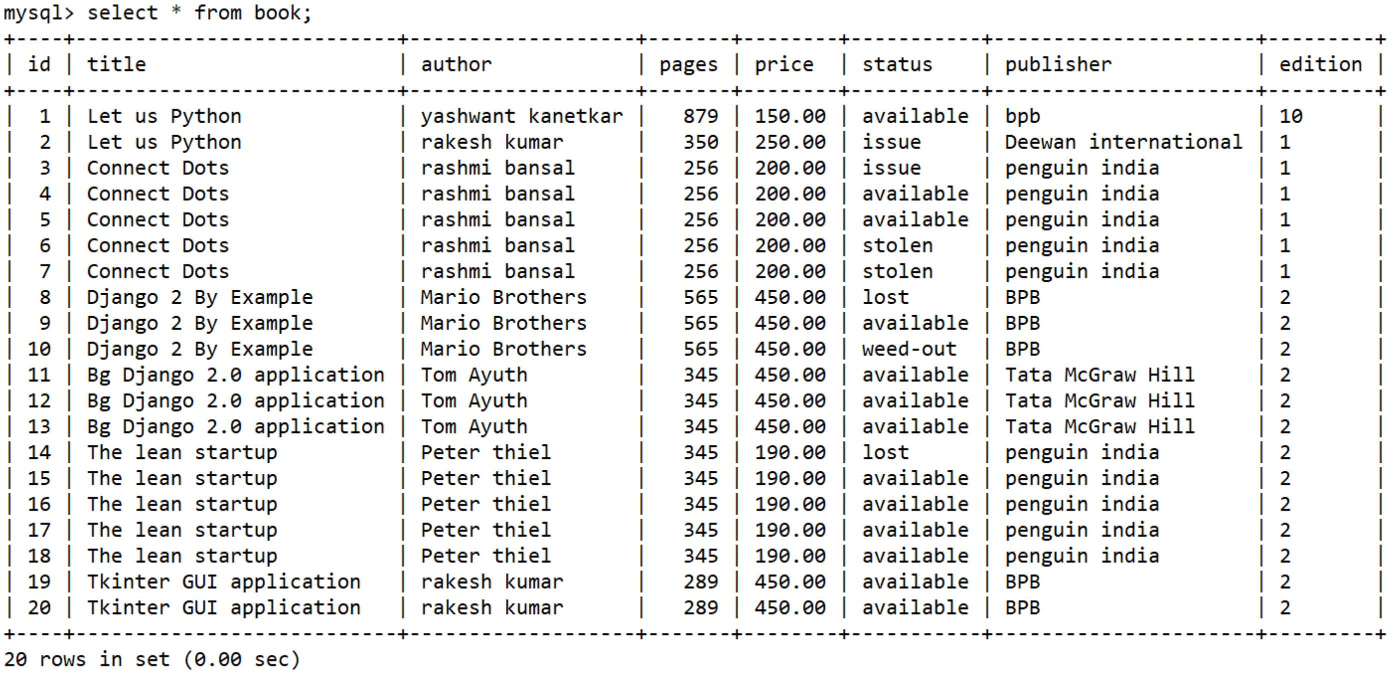


```
mysql> desc member;
+---------+-----------+------+-----+---------+----------------+
| Field   | Type      | Null | Key | Default | Extra          |
+---------+-----------+------+-----+---------+----------------+
| id      | int(11)   | NO   | PRI | NULL    | auto_increment |
| name    | char(30)  | YES  |     | NULL    |                |
| class   | char(15)  | YES  |     | NULL    |                |
| address | char(100) | YES  |     | NULL    |                |
| phone   | char(15)  | YES  |     | NULL    |                |
| email   | char(60)  | YES  |     | NULL    |                |
+---------+-----------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

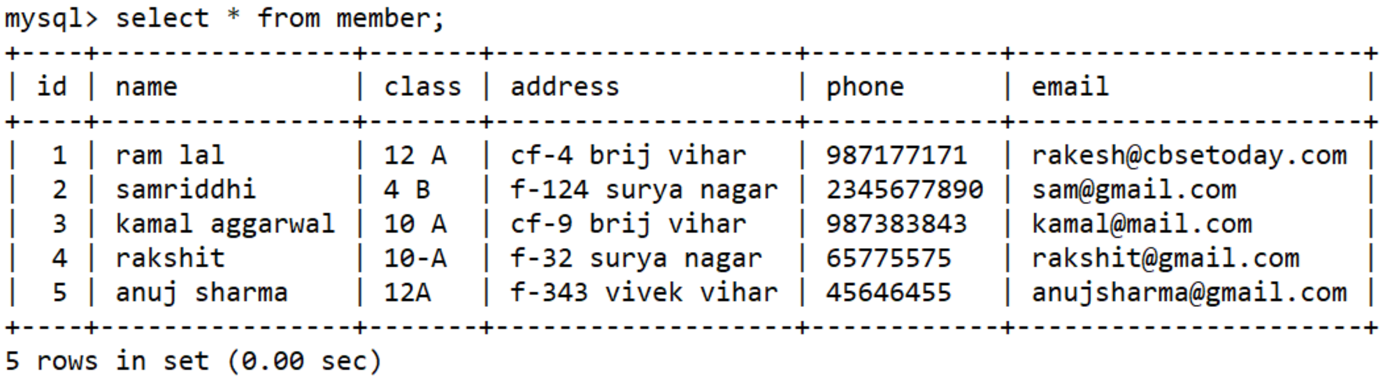


```
mysql> desc transaction;
+-------+------------+------+-----+---------+----------------+
| Field | Type       | Null | Key | Default | Extra          |
+-------+------------+------+-----+---------+----------------+
| tid   | int(11)    | NO   | PRI | NULL    | auto_increment |
| b_id  | int(11)    | YES  |     | NULL    |                |
| m_id  | int(11)    | YES  |     | NULL    |                |
| doi   | date       | YES  |     | NULL    |                |
| dor   | date       | YES  |     | NULL    |                |
| fine  | float(5,2) | YES  |     | NULL    |                |
+-------+------------+------+-----+---------+----------------+
6 rows in set (0.01 sec)
```

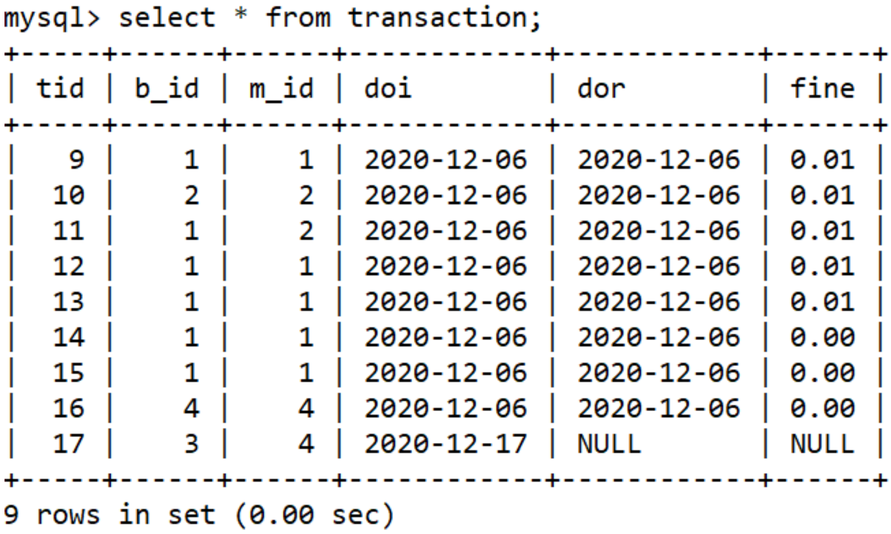
Data in these Tables.



```
mysql> select * from book;
+----+----------------------------+------------------+-------+--------+-----------+-----------------------+---------+
| id | title                      | author           | pages | price  | status    | publisher             | edition |
+----+----------------------------+------------------+-------+--------+-----------+-----------------------+---------+
|  1 | Let us Python              | yashwant kanetkar|   879 | 150.00 | available | bpb                   | 10      |
|  2 | Let us Python              | rakesh kumar     |   350 | 250.00 | issue     | Deewan international  | 1       |
|  3 | Connect Dots               | rashmi bansal    |   256 | 200.00 | issue     | penguin india         | 1       |
|  4 | Connect Dots               | rashmi bansal    |   256 | 200.00 | available | penguin india         | 1       |
|  5 | Connect Dots               | rashmi bansal    |   256 | 200.00 | available | penguin india         | 1       |
|  6 | Connect Dots               | rashmi bansal    |   256 | 200.00 | stolen    | penguin india         | 1       |
|  7 | Connect Dots               | rashmi bansal    |   256 | 200.00 | stolen    | penguin india         | 1       |
|  8 | Django 2 By Example        | Mario Brothers   |   565 | 450.00 | lost      | BPB                   | 2       |
|  9 | Django 2 By Example        | Mario Brothers   |   565 | 450.00 | available | BPB                   | 2       |
| 10 | Django 2 By Example        | Mario Brothers   |   565 | 450.00 | weed-out  | BPB                   | 2       |
| 11 | Bg Django 2.0 application  | Tom Ayuth        |   345 | 450.00 | available | Tata McGraw Hill      | 2       |
| 12 | Bg Django 2.0 application  | Tom Ayuth        |   345 | 450.00 | available | Tata McGraw Hill      | 2       |
| 13 | Bg Django 2.0 application  | Tom Ayuth        |   345 | 450.00 | available | Tata McGraw Hill      | 2       |
| 14 | The lean startup           | Peter thiel      |   345 | 190.00 | lost      | penguin india         | 2       |
| 15 | The lean startup           | Peter thiel      |   345 | 190.00 | available | penguin india         | 2       |
| 16 | The lean startup           | Peter thiel      |   345 | 190.00 | available | penguin india         | 2       |
| 17 | The lean startup           | Peter thiel      |   345 | 190.00 | available | penguin india         | 2       |
| 18 | The lean startup           | Peter thiel      |   345 | 190.00 | available | penguin india         | 2       |
| 19 | Tkinter GUI application    | rakesh kumar     |   289 | 450.00 | available | BPB                   | 2       |
| 20 | Tkinter GUI application    | rakesh kumar     |   289 | 450.00 | available | BPB                   | 2       |
+----+----------------------------+------------------+-------+--------+-----------+-----------------------+---------+
20 rows in set (0.00 sec)
```



```
mysql> select * from member;
+----+----------------+-------+-------------------+------------+-----------------------+
| id | name           | class | address           | phone      | email                 |
+----+----------------+-------+-------------------+------------+-----------------------+
|  1 | ram lal        | 12 A  | cf-4 brij vihar   | 987177171  | rakesh@cbsetoday.com  |
|  2 | samriddhi      | 4 B   | f-124 surya nagar | 2345677890 | sam@gmail.com         |
|  3 | kamal aggarwal | 10 A  | cf-9 brij vihar   | 987383843  | kamal@mail.com        |
|  4 | rakshit        | 10-A  | f-32 surya nagar  | 65775575   | rakshit@gmail.com     |
|  5 | anuj sharma    | 12A   | f-343 vivek vihar | 45646455   | anujsharma@gmail.com  |
+----+----------------+-------+-------------------+------------+-----------------------+
5 rows in set (0.00 sec)
```



```
mysql> select * from transaction;
+-----+------+------+------------+------------+------+
| tid | b_id | m_id | doi        | dor        | fine |
+-----+------+------+------------+------------+------+
|   9 |    1 |    1 | 2020-12-06 | 2020-12-06 | 0.01 |
|  10 |    2 |    2 | 2020-12-06 | 2020-12-06 | 0.01 |
|  11 |    1 |    2 | 2020-12-06 | 2020-12-06 | 0.01 |
|  12 |    1 |    1 | 2020-12-06 | 2020-12-06 | 0.01 |
|  13 |    1 |    1 | 2020-12-06 | 2020-12-06 | 0.01 |
|  14 |    1 |    1 | 2020-12-06 | 2020-12-06 | 0.00 |
|  15 |    1 |    1 | 2020-12-06 | 2020-12-06 | 0.00 |
|  16 |    4 |    4 | 2020-12-06 | 2020-12-06 | 0.00 |
|  17 |    3 |    4 | 2020-12-17 | NULL       | NULL |
+-----+------+------+------------+------------+------+
9 rows in set (0.00 sec)
```

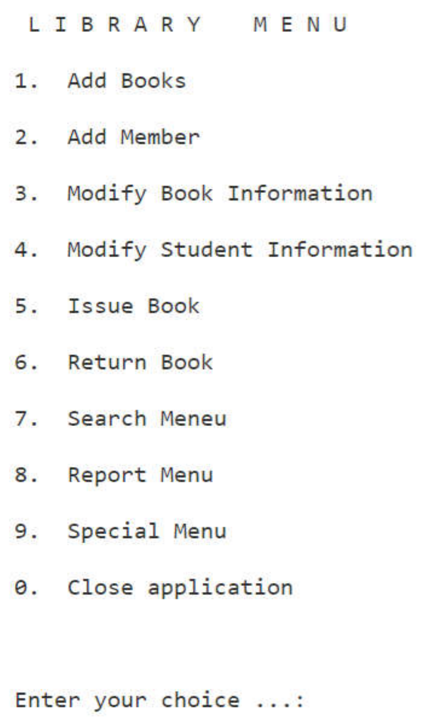
Python Program

```
#   project name    : Library management system

#   made by         : Swarnima and Divya
#   session         : 2020-21
#   roll no         :

import mysql.connector
from datetime import date
from prettytable import PrettyTable

fine_per_day =1.0  #global variable

def clear():
  for _ in range(65):
     print()


def add_book():
  conn = mysql.connector.connect(
       host='localhost', database='library', user='root', password='')
  cursor = conn.cursor()

  title = input('Enter Book Title :')
  author = input('Enter Book Author : ')
  publisher = input('Enter Book Publisher : ')
  pages = input('Enter Book Pages : ')
  price = input('Enter Book Price : ')
  edition = input('Enter Book Edition : ')
  copies  = int(input('Enter copies : '))
  sql = 'insert into book(title,author,price,pages,publisher,edition,status) valu
es ( "' + \
       title + '","' + author+'",'+price+','+pages+',"'+publisher+'","'+edition+'
","available");'
   #sql2 = 'insert into transaction(dot,qty,type) values ("'+str(today)+'",'+qty+
',"purchase");'
  #print(sql)
  for _ in range(0,copies):
    cursor.execute(sql)
  conn.close()
  print('\n\nNew Book added successfully')
  wait = input('\n\n\n Press any key to continue....')


def add_member():
  conn = mysql.connector.connect(
      host='localhost', database='library', user='root', password='')
  cursor = conn.cursor()

  name = input('Enter Member Name :')
  clas = input('Enter Member Class & Section : ')
  address = input('Enter Member Address : ')
  phone = input('Enter Member Phone  : ')
  email = input('Enter Member Email  : ')


  sql = 'insert into member(name,class,address,phone,email) values ( "' + \
      name + '","' + clas+'","'+address+'","'+phone + \
        '","'+email+'");'
  #sql2 = 'insert into transaction(dot,qty,type) values ("'+str(today)+'",'+qty+'
,"purchase");'
  #print(sql)

  cursor.execute(sql)
  conn.close()
  print('\n\nNew Member added successfully')
  wait = input('\n\n\n Press any key to continue....')


def modify_book():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    clear()
    print('Modify BOOK Details Screen ')
    print('-'*120)
    print('\n1. Book Title')
    print('\n2. Book Author')
    print('\n3. Book Publisher')
    print('\n4. Book Pages')
    print('\n5. Book Price')
    print('\n6. Book Edition')
    print('\n\n')
    choice = int(input('Enter your choice :'))
    field = ''
    if choice == 1:
        field = 'title'
    if choice == 2:
        field = 'author'
    if choice == 3:
        field = 'publisher'
    if choice == 4:
        field = 'pages'
    if choice == 5:
        field = 'price'
    book_id = input('Enter Book ID :')
    value = input('Enter new value :')
    if field =='pages' or field == 'price':
        sql = 'update book set ' + field + ' = '+value+' where id = '+book_id+';'
    else:
        sql = 'update book set ' + field + ' = "'+value+'" where id = '+book_id+'
;'
    #print(sql)
    cursor.execute(sql)
    print('\n\n\nBook details Updated.....')
    conn.close()
    wait = input('\n\n\n Press any key to continue....')


def modify_member():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    clear()
    print('Modify Memeber Information Screen ')
    print('-'*120)
    print('\n1. Name')
    print('\n2. Class')
    print('\n3. address')
    print('\n4. Phone')
    print('\n5. Emaile')
    print('\n\n')
    choice = int(input('Enter your choice :'))
    field =''
    if choice == 1:
        field ='name'
    if choice == 2:
        field = 'class'
    if choice ==3:
        field ='address'
    if choice == 4:
        field = 'phone'
    if choice == 5:
        field = 'email'
    mem_id =input('Enter member ID :')
    value = input('Enter new value :')
    sql = 'update member set '+ field +' = "'+value+'" where id = '+mem_id+';'
    #print(sql)
    cursor.execute(sql)
    print('Member details Updated.....')
    conn.close()
    wait = input('\n\n\n Press any key to continue....')


def mem_issue_status(mem_id):
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    sql ='select * from transaction where m_id ='+mem_id +' and dor is NULL;'
    #print(sql)
    cursor.execute(sql)
    results = cursor.fetchall()
    return results


def book_status(book_id):
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    sql = 'select * from book where id ='+book_id + ';'
    cursor.execute(sql)
    result = cursor.fetchone()
    return result[5]

def book_issue_status(book_id,mem_id):
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    sql = 'select * from transaction where b_id ='+book_id + ' and m_id ='+ mem_i
d +' and dor is NULL;'
    cursor.execute(sql)
    result = cursor.fetchone()
    return result

def issue_book():
    conn = mysql.connector.connect(
      host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n BOOK ISSUE SCREEN ')
    print('-'*120)
    book_id = input('Enter Book  ID : ')
    mem_id  = input('Enter Member ID :')


    result = book_status(book_id)
    result1 = mem_issue_status(mem_id)
    #print(result1)
    today = date.today()
    if len(result1) == 0:
      if result == 'available':
          sql = 'insert into transaction(b_id, m_id, doi) values('+book_id+','+me
m_id+',"'+str(today)+'");'
          sql_book = 'update book set status="issue" where id ='+book_id + ';'
          cursor.execute(sql)
          cursor.execute(sql_book)
          print('\n\n\n Book issued successfully')
      else:
          print('\n\nBook is not available for ISSUE... Current status :',result1
)
    else:
      if len(result1)<1:
        sql = 'insert into transaction(b_id, m_id, doi) values(' + \
             book_id+','+mem_id+',"'+str(today)+'");'
        sql_book = 'update book set status="issue" where id ='+book_id + ';'
        #print(len(result))
        cursor.execute(sql)
        cursor.execute(sql_book)
        print('\n\n\n Book issued successfully')
      else:
        print('\n\nMember already have book from the Library')
      #print(result)

    conn.close()
    wait = input('\n\n\n Press any key to continue....')

def return_book():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    global fine_per_day
    clear()
    print('\n BOOK RETURN SCREEN ')
    print('-'*120)
    book_id = input('Enter Book  ID : ')
    mem_id = input('Enter Member ID :')
    today =date.today()
    result = book_issue_status(book_id,mem_id)
    if result==None:
       print('Book was not issued...Check Book Id and Member ID again..')
    else:
       sql='update book set status ="available" where id ='+book_id +';'
       din = (today - result[3]).days
       fine = din * fine_per_day    #  fine per data
       sql1 = 'update transaction set dor ="'+str(today)+'" , fine='+str(fine)+' where b_id='+book_id +' and m_id='+mem_id+' and dor is NULL;'

       cursor.execute(sql)
       cursor.execute(sql1)
       print('\n\nBook returned successfully')
    conn.close()
    wait = input('\n\n\n Press any key to continue....')

def search_book(field):
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n BOOK SEARCH SCREEN ')
    print('-'*120)
    msg ='Enter '+ field +' Value :'
    title = input(msg)
    sql ='select * from book where '+ field + ' like "%'+ title+'%"'
    cursor.execute(sql)
    records = cursor.fetchall()
    clear()
    print('Search Result for :',field,' :' ,title)
    print('-'*120)
    for record in records:
      print(record)
    conn.close()
    wait = input('\n\n\n Press any key to continue....')

def search_menu():
    while True:
      clear()
      print(' S E A R C H   M E N U ')
      print("\n1.  Book Title")
      print('\n2.  Book Author')
      print('\n3.  Publisher')
      print('\n4.  Exit to main Menu')
      print('\n\n')
      choice = int(input('Enter your choice ...: '))
      field =''
      if choice == 1:
        field='title'
      if choice == 2:
        field = 'author'
      if choice == 3:
        field = 'publisher'
      if choice == 4:
        break
      search_book(field)


def reprot_book_list():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - BOOK TITLES ')
    print('-'*120)
    sql ='select * from book'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_issued_books():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - BOOK TITLES - Issued')
    print('-'*120)
    sql = 'select * from book where status = "issue";'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_available_books():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - BOOK TITLES - Available')
    print('-'*120)
    sql = 'select * from book where status = "available";'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_weed_out_books():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - BOOK TITLES - Weed Out')
    print('-'*120)
    sql = 'select * from book where status = "weed-out";'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_stolen_books():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - BOOK TITLES - Stolen')
    print('-'*120)
    sql = 'select * from book where status = "stolen";'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_lost_books():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - BOOK TITLES - lost')
    print('-'*120)
    sql = 'select * from book where status = "lost";'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_member_list():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()

    clear()
    print('\n REPORT - Members List ')
    print('-'*120)
    sql = 'select * from member'
    cursor.execute(sql)
    records = cursor.fetchall()
    for record in records:
       print(record)
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_fine_collection():
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    sql ='select sum(fine) from transaction where dor ="'+str(date.today())+'";'
    cursor.execute(sql)
    result = cursor.fetchone() #always return values in the form of tuple
    clear()
    print('Fine collection')
    print('-'*120)
    print('Total fine collected Today :',result[0])
    print('\n\n\n')
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')


def report_menu():
    while True:
      clear()
      print(' R E P O R T    M E N U ')
      print("\n1.  Book List")
      print('\n2.  Member List')
      print('\n3.  Issued Books')
      print('\n4.  Available Books')
      print('\n5.  Weed out Book')
      print('\n6.  Stolen Book')
      print('\n7.  Lost Book')
      print('\n8.  Fine Collection')
      print('\n9.  Exit to main Menu')
      print('\n\n')
      choice = int(input('Enter your choice ...: '))

      if choice == 1:
        reprot_book_list()
      if choice == 2:
        report_member_list()
      if choice == 3:
        report_issued_books()
      if choice == 4:
        report_available_books()
      if choice == 5:
        report_weed_out_books()
      if choice == 6:
        report_stolen_books()
      if choice == 7:
        report_lost_books()
      if choice == 8:
        report_fine_collection()
      if choice == 9:
        break


def change_book_status(status,book_id):
    conn = mysql.connector.connect(
        host='localhost', database='library', user='root', password='')
    cursor = conn.cursor()
    sql = 'update book set status = "'+status +'" where id ='+book_id + ' and sta
tus ="available"'
    cursor.execute(sql)
    print('Book status changed to ',status)
    print('\n\n\n')
    conn.close()
    wait = input('\n\n\nPress any key to continue.....')
```

```
def special_menu():

    while True:
      clear()
      print(' S P E C I A L     M E N U')
      print("\n1.  Book Stolen")
      print('\n2.  Book Lost')
      print('\n3.  Book Weed out')
      print('\n4.  Return Book')
      print('\n\n')
      choice = int(input('Enter your choice ...: '))
      status=''
      if choice == 1:
         status ='stolen'
      if choice == 2:
         status = 'lost'
      if choice == 3:
         status = 'weed-out'
      if choice == 4:
         break
      book_id = input('Enter book id :')
      change_book_status(status,book_id)


def main_menu():
    while True:
      clear()
      print(' L I B R A R Y    M E N U')
      print("\n1.  Add Books")
      print('\n2.  Add Member')
      print('\n3.  Modify Book Information')
      print('\n4.  Modify Student Information')
      print('\n5.  Issue Book')
      print('\n6.  Return Book')
      print('\n7.  Search Meneu')
      print('\n8.  Report Menu')
      print('\n9.  Special Menu')
      print('\n0.  Close application')
      print('\n\n')
      choice = int(input('Enter your choice ...: '))

      if choice == 1:
        add_book()
      if choice == 2:
        add_member()
      if choice == 3:
        modify_book()
      if choice == 4:
        modify_member()
      if choice == 5:
        issue_book()
      if choice == 6:
        return_book()
      if choice == 7:
        search_menu()
      if choice == 8:
        report_menu()
      if choice == 9:
        special_menu()
      if choice == 0:
        break


if __name__ == "__main__":
    main_menu()
```

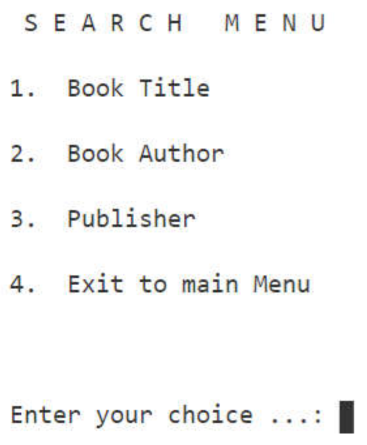
Output Screens

Main screen



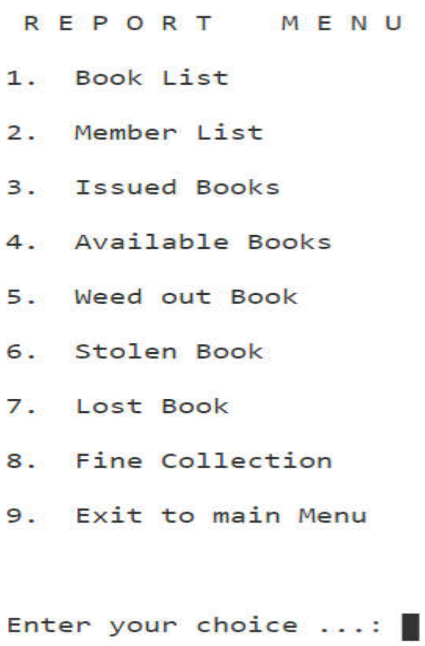
```
  LIBRARY   MENU

1.  Add Books

2.  Add Member

3.  Modify Book Information

4.  Modify Student Information

5.  Issue Book

6.  Return Book

7.  Search Meneu

8.  Report Menu

9.  Special Menu

0.  Close application


Enter your choice ...:
```

Search Menu Screen



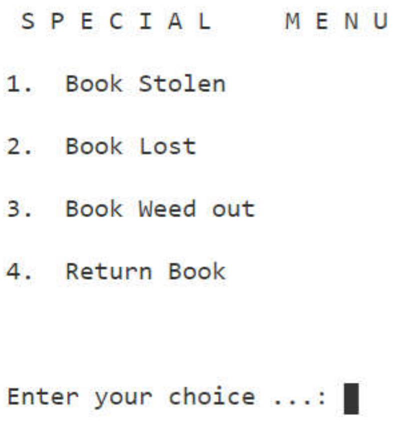
```
  SEARCH   MENU

1.  Book Title

2.  Book Author

3.  Publisher

4.  Exit to main Menu


Enter your choice ...:
```

Report Screen



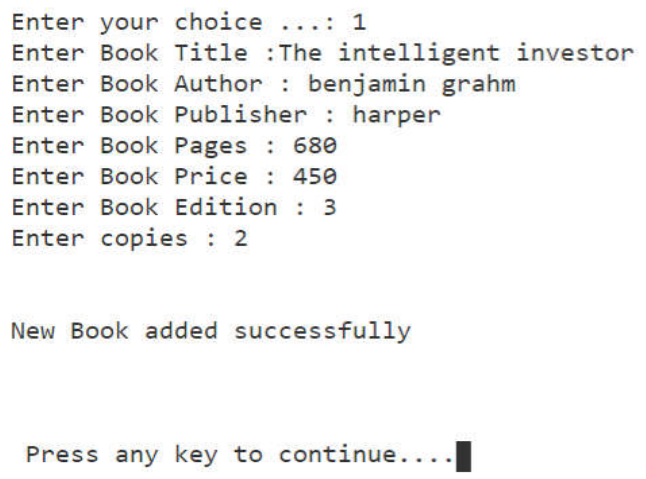
```
  R E P O R T    M E N U

1.   Book List

2.   Member List

3.   Issued Books

4.   Available Books

5.   Weed out Book

6.   Stolen Book

7.   Lost Book

8.   Fine Collection

9.   Exit to main Menu


Enter your choice ...:
```

Special Menu Screen



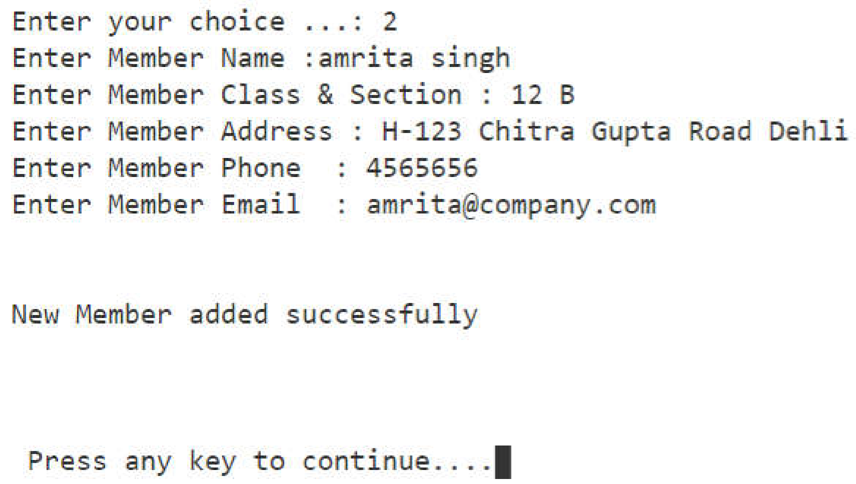
```
  S P E C I A L    M E N U

1.   Book Stolen

2.   Book Lost

3.   Book Weed out

4.   Return Book


Enter your choice ...:
```

Add New Book



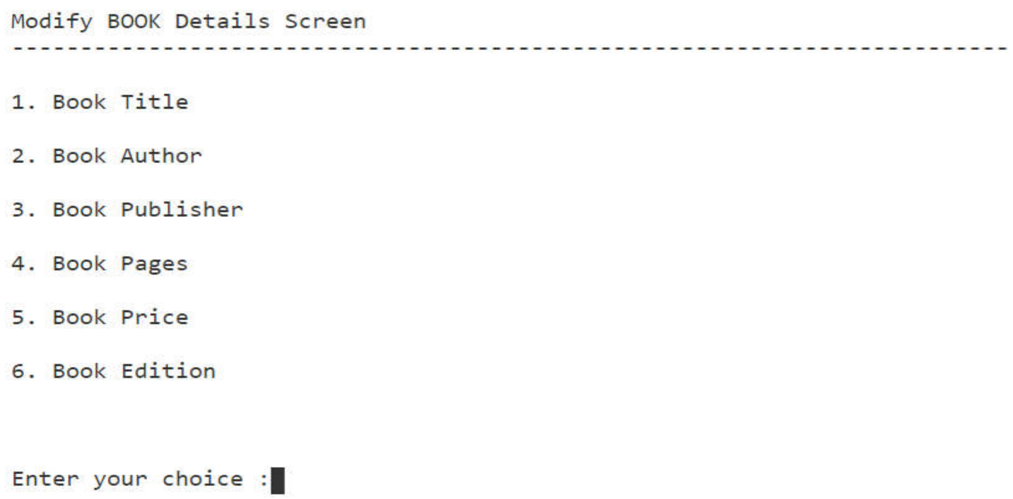
```
Enter your choice ...: 1
Enter Book Title :The intelligent investor
Enter Book Author : benjamin grahm
Enter Book Publisher : harper
Enter Book Pages : 680
Enter Book Price : 450
Enter Book Edition : 3
Enter copies : 2


New Book added successfully



 Press any key to continue....
```

Add A new Member



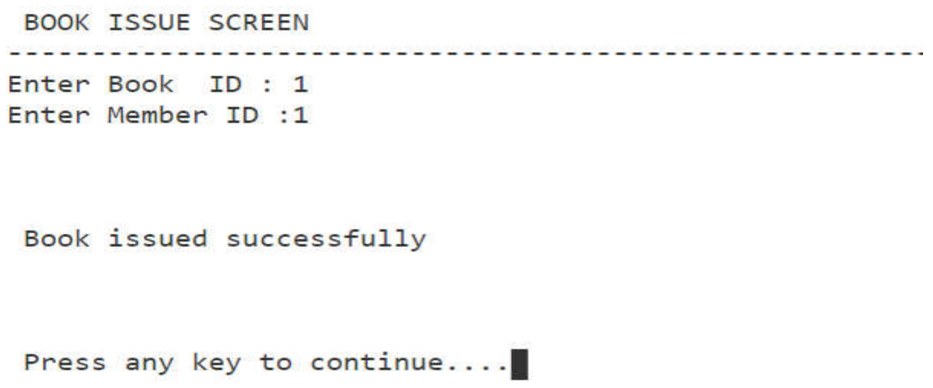
```
Enter your choice ...: 2
Enter Member Name :amrita singh
Enter Member Class & Section : 12 B
Enter Member Address : H-123 Chitra Gupta Road Dehli
Enter Member Phone  : 4565656
Enter Member Email  : amrita@company.com


New Member added successfully



 Press any key to continue....
```

Modify Book Information



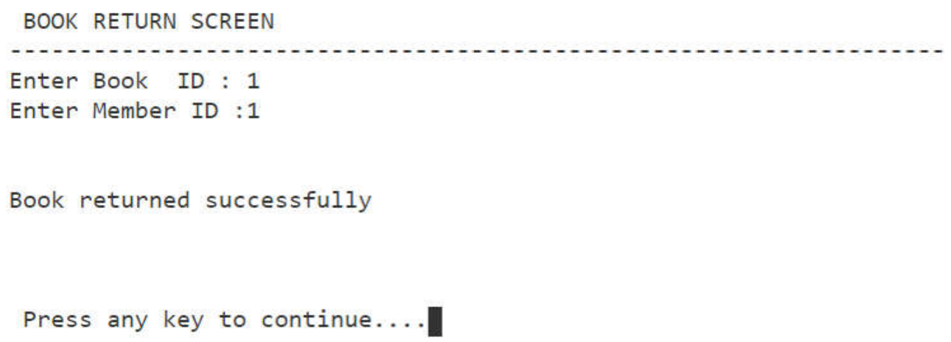
```
Modify BOOK Details Screen
-------------------------------------------------------------------

1. Book Title

2. Book Author

3. Book Publisher

4. Book Pages

5. Book Price

6. Book Edition


Enter your choice :
```

Book Issue Screen



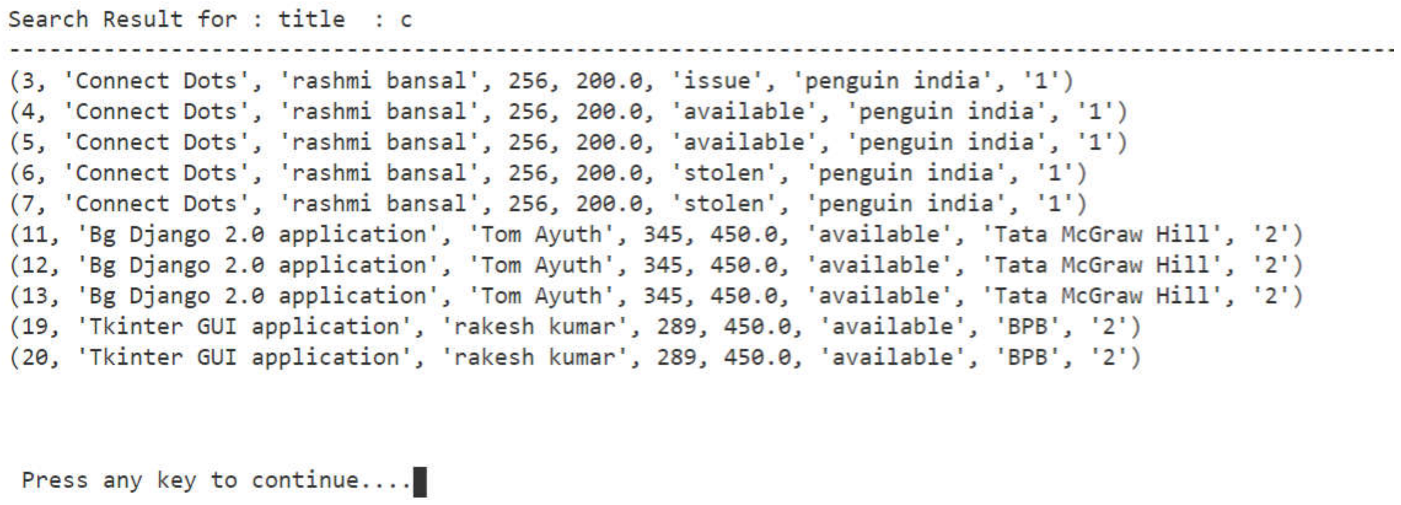
```
  BOOK ISSUE SCREEN
 -------------------------------------------------------
 Enter Book  ID : 1
 Enter Member ID :1



  Book issued successfully



  Press any key to continue....
```

Book Return Screen



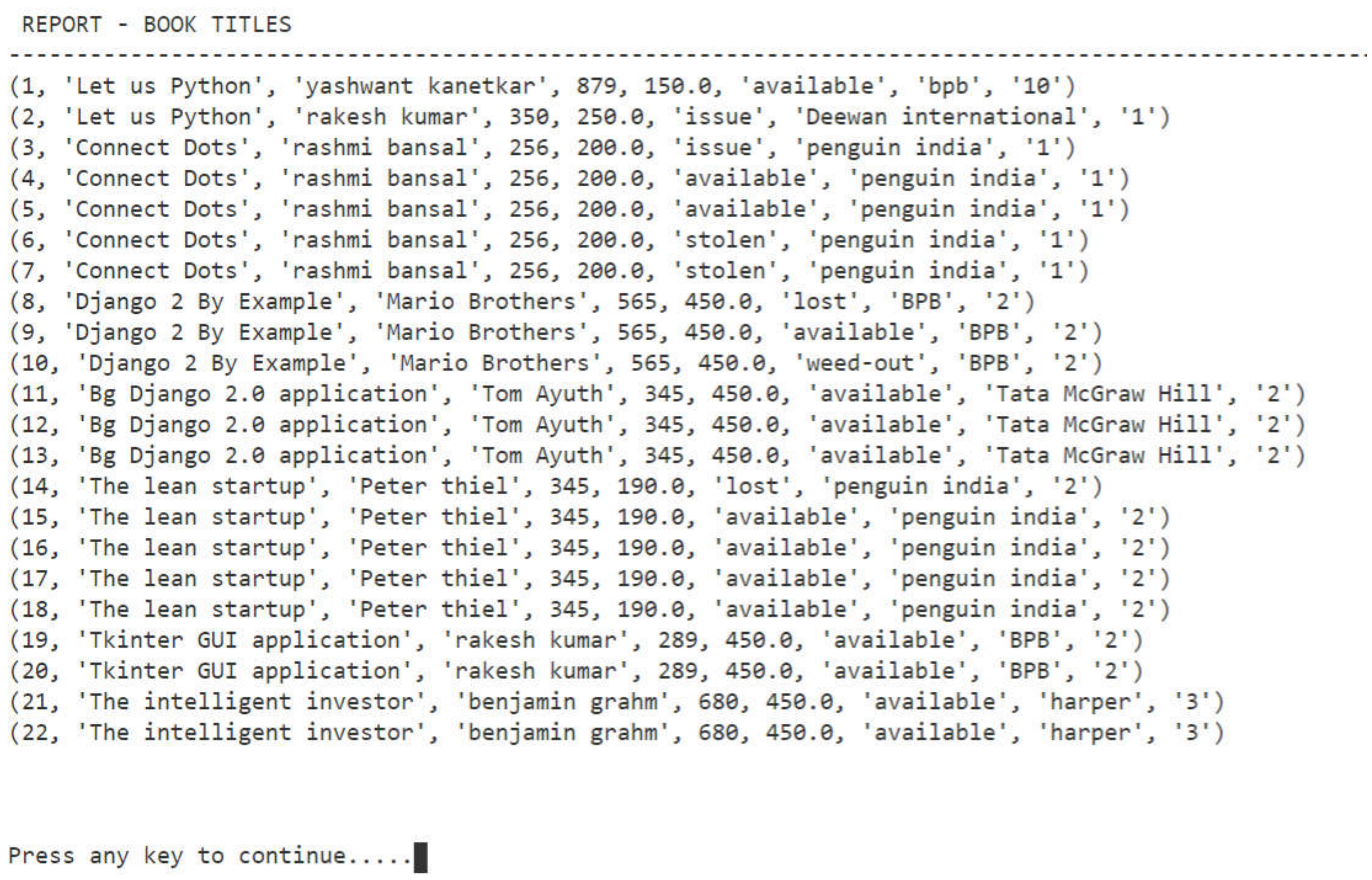
```
  BOOK RETURN SCREEN
 ------------------------------------------------------------------
 Enter Book  ID : 1
 Enter Member ID :1


 Book returned successfully



  Press any key to continue....
```

Search Title



```
 Search Result for : title  : c
 ---------------------------------------------------------------------------------------------------
 (3, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'issue', 'penguin india', '1')
 (4, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'available', 'penguin india', '1')
 (5, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'available', 'penguin india', '1')
 (6, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'stolen', 'penguin india', '1')
 (7, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'stolen', 'penguin india', '1')
 (11, 'Bg Django 2.0 application', 'Tom Ayuth', 345, 450.0, 'available', 'Tata McGraw Hill', '2')
 (12, 'Bg Django 2.0 application', 'Tom Ayuth', 345, 450.0, 'available', 'Tata McGraw Hill', '2')
 (13, 'Bg Django 2.0 application', 'Tom Ayuth', 345, 450.0, 'available', 'Tata McGraw Hill', '2')
 (19, 'Tkinter GUI application', 'rakesh kumar', 289, 450.0, 'available', 'BPB', '2')
 (20, 'Tkinter GUI application', 'rakesh kumar', 289, 450.0, 'available', 'BPB', '2')



  Press any key to continue....
```

Books Report Sample



```
 REPORT - BOOK TITLES
----------------------------------------------------------------------------------------------------
(1, 'Let us Python', 'yashwant kanetkar', 879, 150.0, 'available', 'bpb', '10')
(2, 'Let us Python', 'rakesh kumar', 350, 250.0, 'issue', 'Deewan international', '1')
(3, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'issue', 'penguin india', '1')
(4, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'available', 'penguin india', '1')
(5, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'available', 'penguin india', '1')
(6, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'stolen', 'penguin india', '1')
(7, 'Connect Dots', 'rashmi bansal', 256, 200.0, 'stolen', 'penguin india', '1')
(8, 'Django 2 By Example', 'Mario Brothers', 565, 450.0, 'lost', 'BPB', '2')
(9, 'Django 2 By Example', 'Mario Brothers', 565, 450.0, 'available', 'BPB', '2')
(10, 'Django 2 By Example', 'Mario Brothers', 565, 450.0, 'weed-out', 'BPB', '2')
(11, 'Bg Django 2.0 application', 'Tom Ayuth', 345, 450.0, 'available', 'Tata McGraw Hill', '2')
(12, 'Bg Django 2.0 application', 'Tom Ayuth', 345, 450.0, 'available', 'Tata McGraw Hill', '2')
(13, 'Bg Django 2.0 application', 'Tom Ayuth', 345, 450.0, 'available', 'Tata McGraw Hill', '2')
(14, 'The lean startup', 'Peter thiel', 345, 190.0, 'lost', 'penguin india', '2')
(15, 'The lean startup', 'Peter thiel', 345, 190.0, 'available', 'penguin india', '2')
(16, 'The lean startup', 'Peter thiel', 345, 190.0, 'available', 'penguin india', '2')
(17, 'The lean startup', 'Peter thiel', 345, 190.0, 'available', 'penguin india', '2')
(18, 'The lean startup', 'Peter thiel', 345, 190.0, 'available', 'penguin india', '2')
(19, 'Tkinter GUI application', 'rakesh kumar', 289, 450.0, 'available', 'BPB', '2')
(20, 'Tkinter GUI application', 'rakesh kumar', 289, 450.0, 'available', 'BPB', '2')
(21, 'The intelligent investor', 'benjamin grahm', 680, 450.0, 'available', 'harper', '3')
(22, 'The intelligent investor', 'benjamin grahm', 680, 450.0, 'available', 'harper', '3')



Press any key to continue.....
```

Reference

1. Computer Projects
    1. By – Rakesh Kumar

2. Computer science
    1. By Sumita Arora

3. The art and Science of Computer Project
    1. SitePoint
4. Online reference
    1. www.pscode.com
    2. www.about.com
    3. www.programmersheaven.com
    4. www.class-12-computer-project.blogspot.in