**PROJECT ON**

**LIBRARY MANAGEMENT SYSTEM**

Submitted By:

Brahmaiah Rachapudi

Abhishek Kukku

Sudhamsa Nagala

**INDEX**

|  |  |
| --- | --- |
| Sr.no | Particulars |
| 1. | Introduction |
| 2. | Project Analysis |
| 3. | Functions And Modules |
| 4. | System Design |
| 5. | Source code |
| 6. | Outputs and Tables |

INTRODUCTION

The aims and objectives are as follows:

* The main aim in library management is adding, issuing and returing of books,and also generating reports on it.

Adding books:

Here we add new books into our database.

Issuing books:

Here we issue books to the candidates.

Returning books:

We get books return from candidates.

Delete books:

We delete books here.

Displaying books:

Here we track all the book details.

Reports:

Here we get information about issued and returned books.

PROJECT ANALYSIS

OPERATION ENVIRONMENT :

|  |  |
| --- | --- |
| FRONT END | PYTHON |
| DATA BASE | MYSQL |

Here we have created a online data base:

To connect to database we have to give,

|  |  |
| --- | --- |
| Host | [sql12.freesqldatabase.com](http://sql12.freesqldatabase.com/) |
| Database name | sql12621331 |
| Database user | sql12621331 |
| **Database password** | 1ssdLqTHym |
| **Port number** | 3306 |

FUNCTIONS AND MODULES:

Modules:

* **Import mysql.connector:**

By importing this module,we are able to get the connection between SQL and python.

Functions:

Connect():

It establishes the connection between Python and MYSQL.

Cursor():

It facilitates the row-by-row processing of records in the result set.

Syntax: <cursor object>=<connection object>.cursor()

Execute():

It is used to execute the sql query and get records using python.

Syntax:<cursor object>.execute<sql query string>

Def():

Is a block of code which only runs when it is called.

Fetchall():

Here it will return all the rows from the result set in the form of a tuple containing the records.

Fetchone():

It will return one row from the result set in the form of a tuple containing the records.

Commit():

It provides changes in the database physically.

SYSTEM DESIGN

Table Design:

-Our project has 3 MYSQL tables:-

Book Table for keeping track of book

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Default | key |
| bname | Varchar(100) | Not null |  |
| author | Varchar(100) |  |  |
| bcode | Varchar(10) | Not null | Primary key |
| total | Int |  |  |
| Subject | Varchar(50) |  |  |

Issue table to issue a book

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data type | Default | Key |
| name | Varchar(100) |  |  |
| regno | Int | Not null | Primary key |
| bcode | Varchar(10) | Not null |  |
| issue\_date | Datetime |  |  |

Return\_books table for returning a book

|  |  |  |  |
| --- | --- | --- | --- |
| field | Data type | Default | Key |
| name | Varchar(100) |  |  |
| regno | Int | Not null | Primary key |
| bcode | Varchar(10) | Not null |  |
| Return\_date | Datetime |  |  |

Data Flow Diagram:

**ENTERS SYSTEM**

**PERFORMS TASK**

**USER**

**CHECK VALIDATION**

**DATABASE**

**SELECTS TASK**

IMPLEMENTATION

**Source code:**

import mysql.connector as a

conn = a.connect(host='sql12.freesqldatabase.com',user='sql12621331',password='1ssdLqTHym',database='sql12621331')

my\_cursor=conn.cursor()

conn.commit()

print('connection succesfully created')

def addbook():

book\_name = input('enter book name: ')

book\_author = input('enter author name: ')

book\_code = input('enter book code: ')

total\_books = int(input('total books: '))

subject = input('enter subject: ')

data = (book\_name, book\_author, book\_code, total\_books, subject)

sql = 'insert into books values (%s, %s, %s, %s, %s);'

my\_cursor = conn.cursor()

my\_cursor.execute(sql, data)

conn.commit()

print('book added successfully')

wait = input('press enter to continue')

main()

def issuebook():

sname = input('enter student name: ')

reg\_no = int(input('enter reg no: '))

book\_code = (input('enter book code: '))

issue\_date = input('enter date: ')

data = (sname, reg\_no, book\_code, issue\_date)

sql = 'insert into issue values (%s, %s, %s, %s);'

my\_cursor = conn.cursor()

my\_cursor.execute(sql, data)

conn.commit()

print('book issued successfully to:', sname)

wait = input('press enter to continue')

bookupdate(book\_code, -1)

main()

def returnbook():

sname = input('Enter student name: ')

reg\_no = int(input('Enter reg no: '))

book\_code = (input('Enter book code: '))

return\_date = input('Enter date: ')

data = (sname, reg\_no,book\_code, return\_date)

sql = 'insert into return\_books values (%s, %s, %s, %s);'

my\_cursor = conn.cursor()

my\_cursor.execute(sql, data)

conn.commit()

print('book returned by:',sname)

wait = input('press enter to continue')

bookupdate(book\_code, 1)

def bookupdate(book\_code, update):

sql\_select = 'select total from books WHERE bcode = %s;'

data = (book\_code,)

my\_cursor = conn.cursor()

my\_cursor.execute(sql\_select, data)

myresult = my\_cursor.fetchone()

t = myresult[0] + update

sql\_update = 'update books SET total = %s WHERE bcode = %s;'

data=(t,book\_code)

my\_cursor.execute(sql\_update, data)

conn.commit()

wait = input('press enter to continue')

main()

def deletebook():

book\_code = int(input('enter book code: '))

sql = 'delete from books WHERE bcode = %s;'

data = (book\_code,)

my\_cursor = conn.cursor()

my\_cursor.execute(sql, data)

conn.commit()

print('book deleted successfully')

wait = input('press enter to continue')

main()

def displaybook():

sql = 'select \* from books;'

my\_cursor = conn.cursor()

my\_cursor.execute(sql)

myresult = my\_cursor.fetchall()

for i in myresult:

print('book\_name:', i[0])

print('book\_author:', i[1])

print('book\_code:', i[2])

print('total\_books:', i[3])

print('subject:', i[4])

wait = input('press enter to continue')

main()

def report\_issued\_books():

sql = 'select \* from issue;'

my\_cursor = conn.cursor()

my\_cursor.execute(sql)

myresult = my\_cursor.fetchall()

for i in myresult:

print(myresult)

wait = input('press enter to continue')

main()

def report\_return\_books():

sql = 'select \* from return\_books;'

my\_cursor = conn.cursor()

my\_cursor.execute(sql)

myresult = my\_cursor.fetchall()

for i in myresult:

print(myresult)

wait = input('press enter to continue')

main()

def main():

print('''

LIBRARY MANAGEMENT SYSTEM APPLICATION

1. ADD BOOK

2. ISSUE BOOK

3. RETURN BOOK

4. DELETE BOOK

5. DISPLAY BOOKS

6. REPORT MENU

7. EXIT PROGRAM

''')

choice = input('enter task no: ')

if choice == '1':

addbook()

elif choice == '2':

issuebook()

elif choice == '3':

returnbook()

elif choice == '4':

deletebook()

elif choice == '5':

displaybook()

elif choice == '6':

print('''REPORT MENU

1. ISSUED BOOKS

2. RETURNED BOOKS

3. GO BACK TO MAIN MENU

''')

choice = input('enter task no: ')

if choice == '1':

report\_issued\_books()

elif choice == '2':

report\_return\_books()

elif choice == '3':

main()

else:

print('please try again')

main()

elif choice == '7':

print('Thank you and have a great day ahead')

else:

print('please try again')

main()

main()

System Testing

Here we perform dry testing,

The aim of the system testing process is to determine all defects in our project.

Unit Testing:-

Here we check complete environment i.e. by importing module whether the sql connection has established or not.

Integration Testing:-

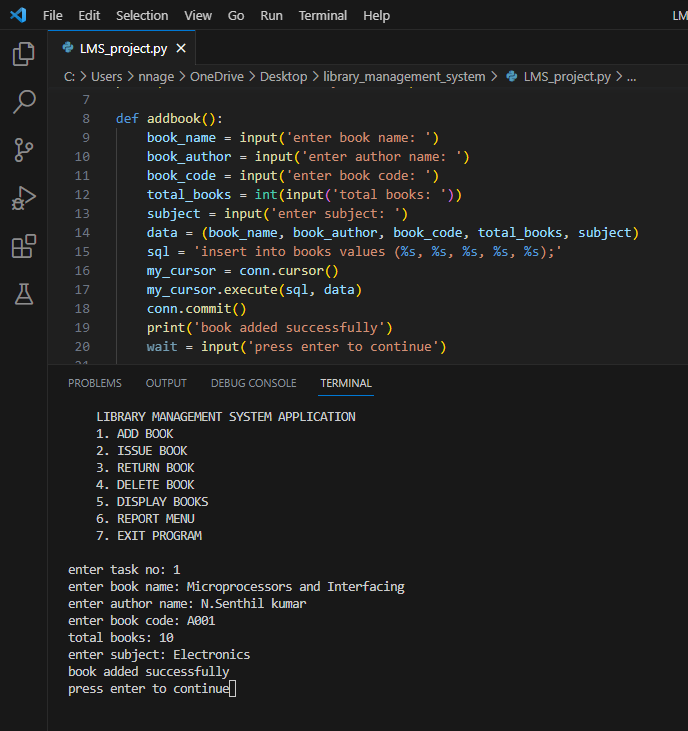
Here we check whether we are getting expected outputs are not.

* To add book, it will check the book code should be unique value.
* To issue book, before it will validate the student details like student registration number should be unique value.
* Return book, here validates the book code and student registration number
* Report on books, validates whether the system updating about book issued and book returned details.

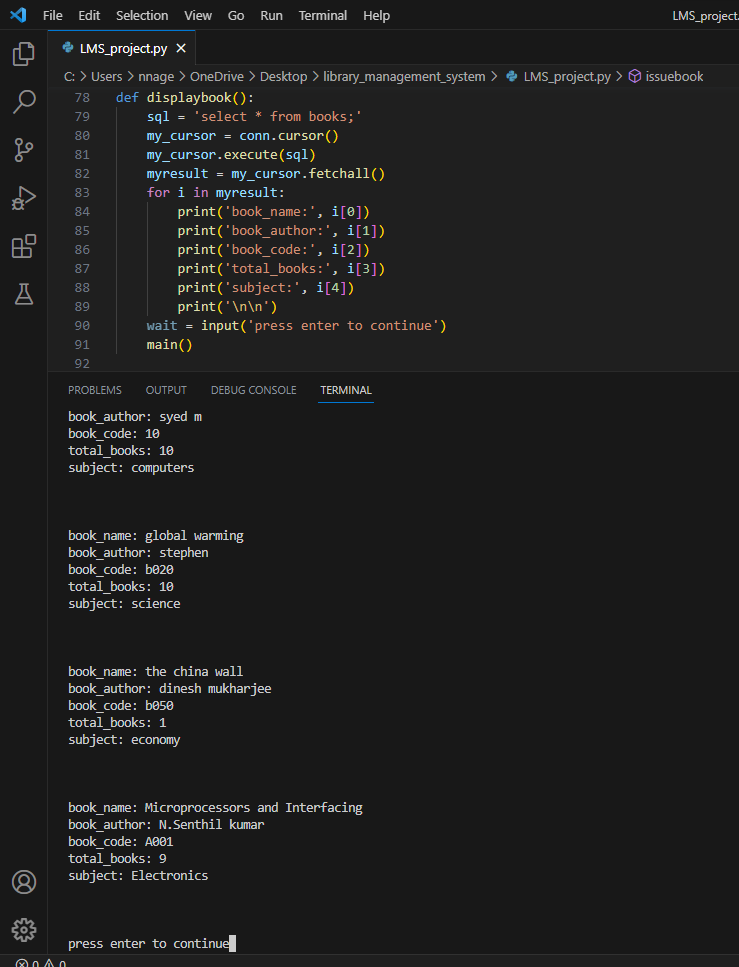
We user selects other than the mention tasks it will display as “invalid please try again”.

OUTPUTS AND TABLES

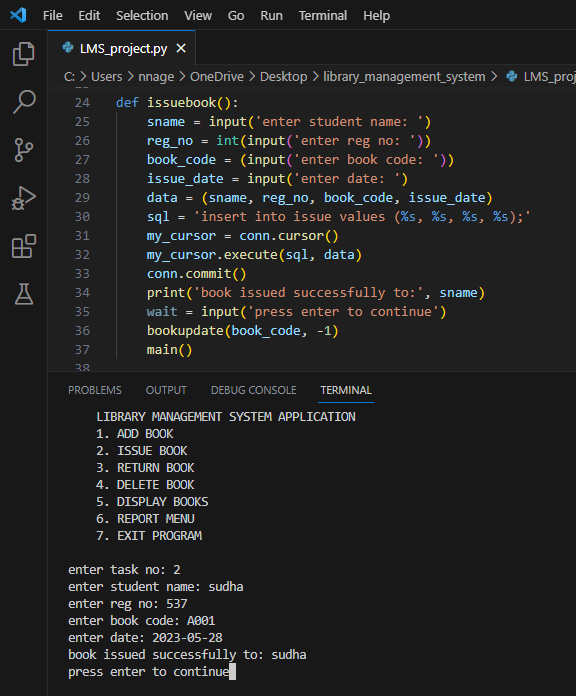
Add Book:



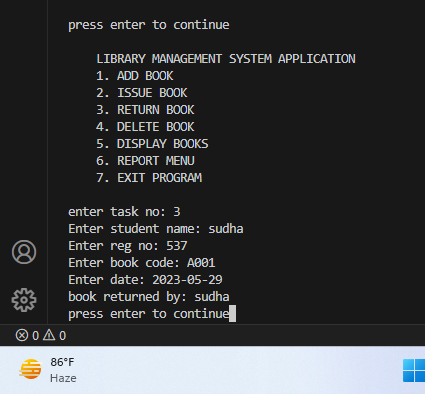
Display Books:



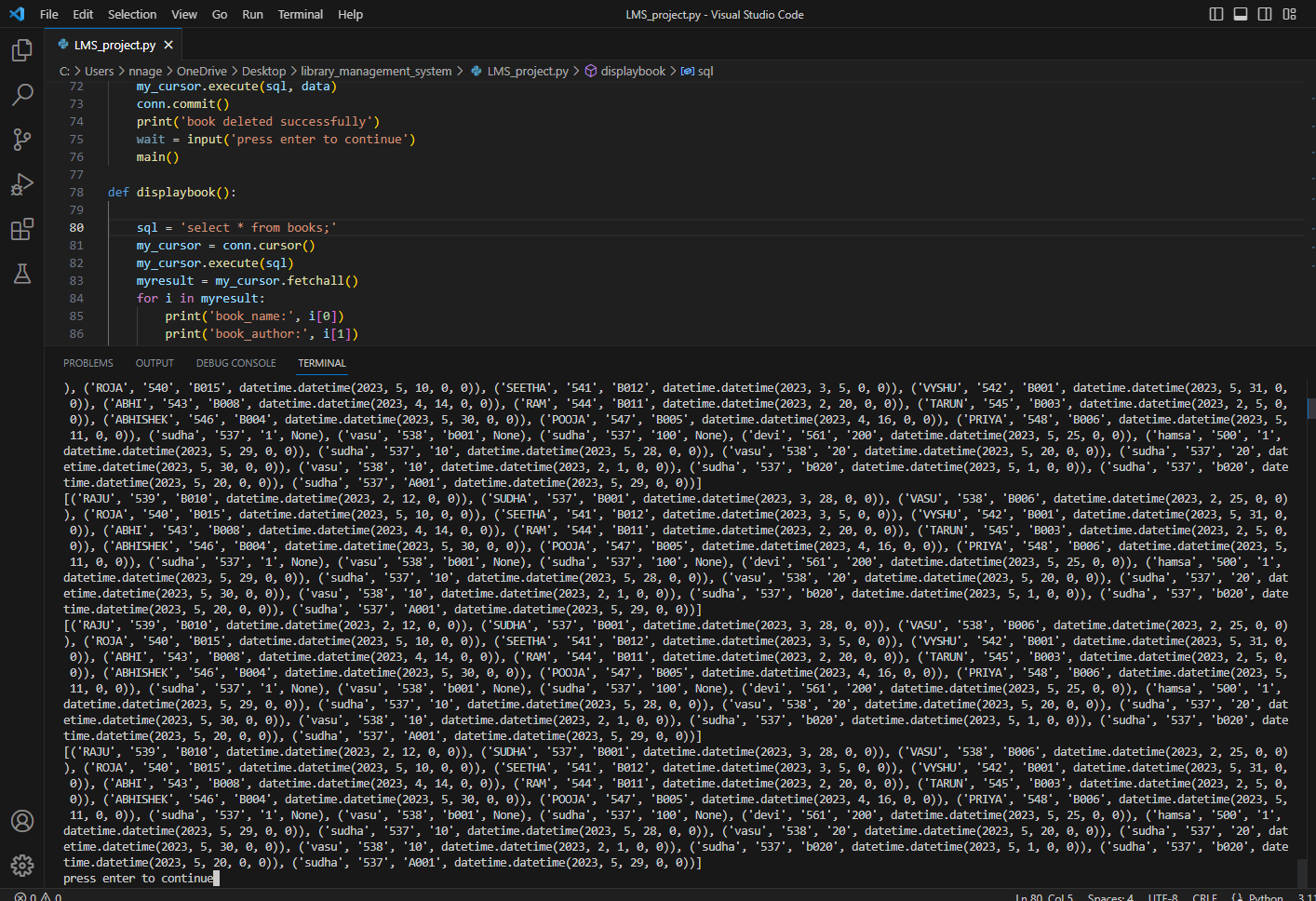
Issue Book:



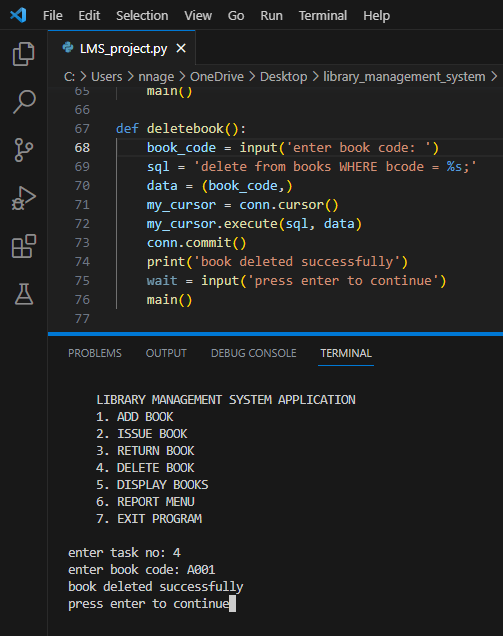
Return Book:



Returned Book\_Report:

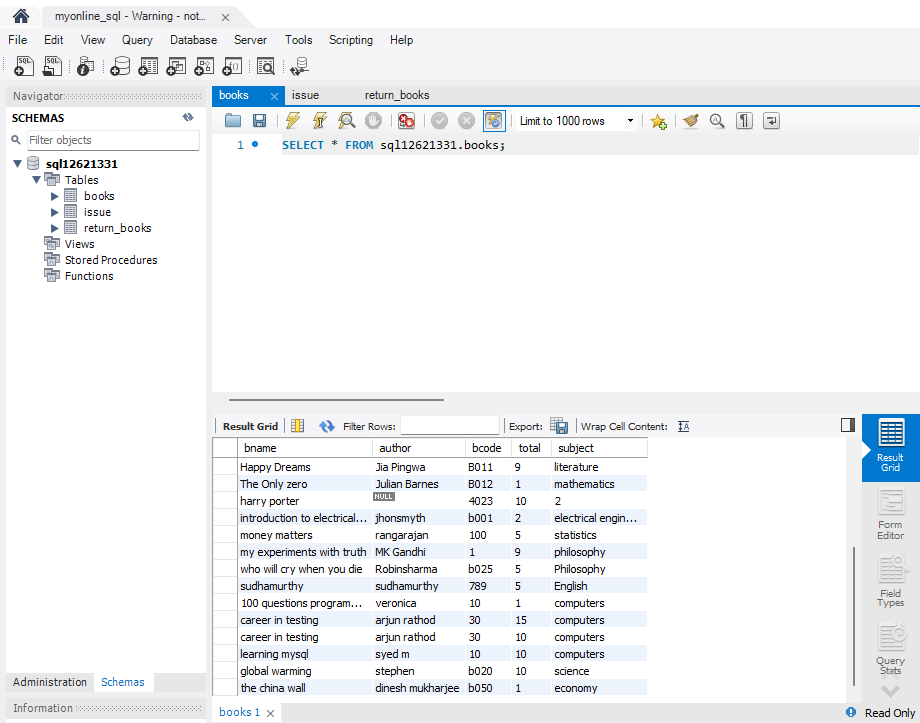


Delete Book:

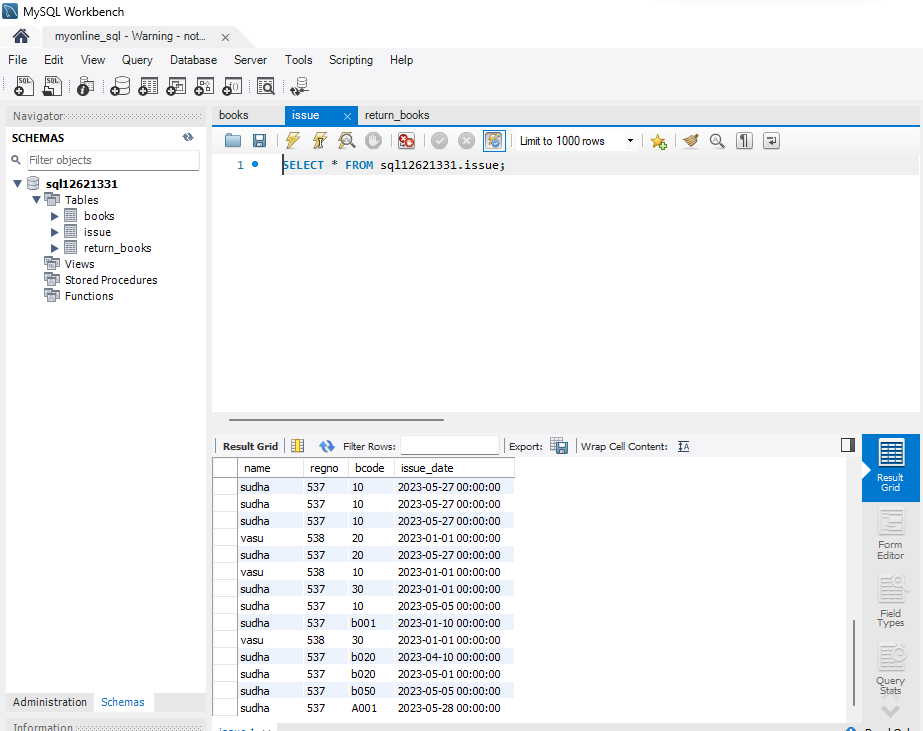


SQL Tables:

Book Table:



Issue Table:



Return Table:

