Book Management System

Version 1.0 (2020-2021)

Computer Science (083) Project

Developed By

Aryamun Narayan Das, XII-F, E09514

Utkarsh Lal, XII-F, R20170

Delhi Public School, R.K.Puram, New Delhi

www.dpsrkp.net

Index

Certificate

This is to certify that _Python Book Management System_ Computer Science project is developed by **_Utkarsh Lal** and **_Aryamun Das_** under my supervision in the session 2020-2021.

The work done by them is original.

Shalini Harisukh
Computer Science Teacher
Date: _3/18/2021_

Acknowledgement

I would like to express my sincere gratitude to my computer teacher Ms.Shalini Harisukh for her vital support, guidance and encouragement without which this project would not come forth from my side. Who helped me complete the project by giving ideas, thoughts and made this project easy and accurate.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way, without which I would be unable to complete my project.

References

1. **Class notes**
2. **NCERT COMPUTER SCIENCE CLASS 12 TEXTBOOKS**

Introduction

Keeping in view of growing requirements of effective and efficient solutions for various problems, with our knowledge of Python and SQL, in this project, we have tried to provide a simple solution for the Library management.

Our project has the complete management solutions to take care of all the day to day issuing and return of books from the library.

We have divided the program into small user defined functions to take care of all future upgrades and expansion. The following is the list of Functions along with their description.

Sno	Function Name	Description
1	Add_Book()	To add new book in the library and new student in the student database
2	View()	To display both tables
3	Searchbook()	Search a record in the database and returns the status of the record
4	IssueBook()	To issue a book
5	Studentnumbers()	Display the number of students
6	returned()	To return a book

Tables with structure used in the program

Table Name: BOOKS

Sno	FieldName/ Column Name	Data Type	Description
1	BCODE	INT NOT NULL	Primary key of the table
2	BNAME	VARCHAR	Book name
3	BAUTH	VARCHAR	Book author
4	SCODE	INT NOT NULL	Foreign key
5	STATUS	VARCHAR	Status of the book
6	DATEOFISSUE	DATE	Date issued(if issued)
7	DATEDUE	DATE	Date due(if issued)

Table Name: STUDENT

Sno	FieldName/ Column Name	Data Type	Description
1	SCODE	INT NOT NULL	Primary key
2	SNAME	VARCHAR	Student name
3	BCODE	INT NOT NULL	Foreign key

External Packages/Modules used in the Program/Software

pymysql - Pure Python MySQL client. According to the maintainer of both mysqlclient and PyMySQL. The user operates databases within the python shell, however, changes need to be committed to the host SQL server, of which the python shell is an instance.

Salient Features of the Project

- User friendly menu driven options
- Exceptions used for the databases and book management to avoid inconvenience to users
- Data Validation to avoid ambiguous data entry

Source Code

```
# Project Title  : Book Management System
# Version        : 1.0 2020-2021
# Developed By   : Aryamun Das, Utkarsh Lal
# Guide          : Ms Shalini Harisukh
# Last Updated On: January 2021
```

```
import MySQLdb #import the module, case sensitive
db=MySQLdb.connect(host=’localhost’, user=’Root’, password=”sql“,
port=3306) #login to your database
cursor=db.cursor() #creates a command prompt variable ‘cursor’
try:
   cursor.execute(‘CREATE DATABASE LIBRARY’)
   print(‘Created library successfully’)
except:
   print(‘The database exists’)

cursor.execute(‘USE LIBRARY’)

try:
  cursor.execute(‘CREATE TABLE STUDENT(SCODE INT NOT NULL, SNAME\
  VARCHAR(25) NOT NULL, BCODE INT(3) NOT NULL’))
  print(‘created table, add items now’)

except:
  print(‘table exists’)

try:
  cursor.execute(‘CREATE TABLE BOOKS(BCODE INT NOT NULL, BNAME\
  VARCHAR(25) NOT NULL, BAUTH VARCHAR(25), SCODE INT NOT NULL,STATUS\
  VARCHAR(25) DATEOFISSUE DATE,\
  DATEDUE DATE’))
  print(‘created table, add items now’)

except:
  print(‘table exists’)

def AddBook():
  def NotNull(x):
    if type(x)==None:
        x=input(‘re-enter %g’)%x

  BCODE=input("bcode :")
  NotNull(BCODE)
  BNAME=input("BNAME :")
  NotNull(BNAME)
  BAUTH=input("AUTHOR :")
  NotNull(BAUTH)
  SCODE=input("SCODE:")
  STATUS=input("Status")
  DATEOFISSUE=input("DATE OF ISSUE[YYYY-MM-DD]:")
  DATEDUE=input("DUE DATE [YYYY-MM-DD]")
  SNAME=input("SNAME :")
  SQL="INSERT INTO STUDENT VALUES ("\
  +SCODE+",'"+SNAME+"',"+\
  BCODE+")
  cursor.execute(SQL)
  db.commit()
  SQL="INSERT INTO BOOKS VALUES ("\
  +BCODE+",'"+BNAME+"',"+BAUTH+","+\
  SCODE+",'"+STATUS+"',”+DATEOFISSUE+”,”+DATEDUE+”)"
  cursor.execute(SQL)
  db.commit()
```

```
def View():
  ask=input(‘which table do you want to see?’).upper()
  while ask not in [‘STUDENT’,’BOOKS’]:
   print(‘no such table’)
   ask=input(‘which table do you want to see?’).upper()
  SQL="SELECT * FROM %s" %ask
  MyCur.execute(SQL)
  R=MyCur.fetchall()
  if ask==’STUDENT’:
    for SCODE, SNAME, BCODE in R:
       print("%4d | %20s | %5d | "%\
            (SCODE,SNAME.ljust(20,"_"),BCODE))
  else:
    for BCODE, BNAME, BAUTH, SCODE, STATUS,DATEOFISSUE, DATEDUE in R:
       print("%4d | %20s | %5d | %8.2f | %12s | %2s| %4s |"%\
            (BCODE, BNAME.ljust(20,"_"),BAUTH,SCODE,STATUS,
            DATEOFISSUE,DATEDUE))


def SearchBook():
  Bookno=input("Book code")
  SQL="SELECT * FROM STUDENT WHERE BCODE="+BCODE+""
  N=MyCur.execute(SQL)
  if N>0:
  R=MyCur.fetchone()
  print("Found...")
  print("BCODE :",R[0])
  print("BNAME :",R[1])
  print("AUTHOR :",R[2])
  print("SCODE :",R[3])
  print(‘STATUS’ R[4])
  if R[4]=’Issued’:
     print(‘Book Issued’)

def IssueBook():
 BCODE=input("BCODE:")
 SQL="SELECT * FROM BOOKS WHERE BCODE="+BCODE+""
 N=MyCur.execute(SQL)
 if N>0:
   SCODE=input("CODE OF ISSUE STUDENT:")
   STATUS="ISSUED"
    DATEOFISSUE=input("DATE OF ISSUE[YYYY-MM-DD]:")
   DATEDUE=input("DUE DATE [YYYY-MM-DD]")
   SQL="UPDATE BOOKS SET Qty=Qty+"+Qty\
   +",STATUS='"+STATUS\
   +"',DATEOFISSUE=”+DATEOFISSUE+”,DATEDUE=”+DATEDUE+”\
    WHERE BCODE="+BCODE+""
   MyCur.execute(SQL)
   db.commit()

def studentnumbers():
  print(“number of students”)
  x=cursor.rowcount()
  print(x)

def returned():
 BCODE=input("BCODE:")
 SCODE=input(“SCODE:”)
 SQL="SELECT * FROM BOOKS WHERE BCODE="+BCODE+"" AND SCODE=”+SCODE””
 N=MyCur.execute(SQL)
 if N>0:
   SCODE=0
   STATUS="Available"
    DATEOFISSUE=None
   DATEDUE=None
   SQL="UPDATE BOOKS SET Qty=Qty+"+Qty\
   +",STATUS='"+STATUS\
   +"',DATEOFISSUE=”+DATEOFISSUE+”,DATEDUE=”+DATEDUE+”\
    WHERE BCODE="+BCODE+""
   MyCur.execute(SQL)
   db.commit()


print("Book Management System 1.0")
while True:
 CH=input("A:Add Item V:View S:Search I:Issue R:Return Q:Quit")
 if CH=='A':
  AddBook()
 elif CH=="S":
  SearchBook()
 elif CH=="V":
  View()
 elif CH=="I":
  IssueBook()
 elif CH=='R':
  returned()
 else:
  break
studentnumbers()
db.close()
```

Output Screen

(All Operations)

```
Book Management System 1.0
A:Add Item V:View S:Search I:Issue R:Return Q:QuitV
A:Add Item V:View S:Search I:Issue R:Return Q:QuitA
Bcode:1
BNAME:NCERT
AUTHOR :CBSE
SCODE=100
STATUS=ISSUED
DATE OF ISSUE[YYYY-MM-DD]:2020-01-01
DUE DATE [YYYY-MM-DD]:2020-01-08
A:Add Item V:View S:Search I:Issue R:Return Q:QuitA
Bcode:2
BNAME:NCERTCHEM
AUTHOR :CBSEMATHS
SCODE=101
STATUS=ISSUED
DATE OF ISSUE[YYYY-MM-DD]:2020-02-01
DUE DATE [YYYY-MM-DD]:2020-02-08
A:Add Item V:View S:Search I:Issue R:Return Q:QuitV
which table do you want to see?books
1| NCERT | CBSE | 100 | ISSUED | 2020-01-01| 2020-01-08  |
2| NCERTCHEM | CBSEMATHS | 101 | ISSUED | 2020-02-01| 2020-02-08  |
A:Add Item V:View S:Search I:Issue R:Return Q:QuitS
Book code2
Found...
BCODE : 2
BNAME : NCERTCHEM
AUTHOR : CBSEMATHS
SCODE : 101
STATUS Issued
Book Issued
A:Add Item V:View S:Search I:Issue R:Return Q:QuitR
BCODE: 2
SCODE: 101
A:Add Item V:View S:Search I:Issue R:Return Q:QuitV
2| NCERTCHEM | CBSEMATHS | 0 | AVAILABLE|       |         |
A:Add Item V:View S:Search I:Issue R:Return Q:QuitI
BCODE: 2
SCODE: 101
Bcode:2
DATE OF ISSUE[YYYY-MM-DD]:2020-02-01
DUE DATE [YYYY-MM-DD]:2020-02-08
A:Add Item V:View S:Search I:Issue R:Return Q:QuitV
2| NCERTCHEM | CBSEMATHS | 101 | ISSUED | 2020-02-01| 2020-02-08  |
A:Add Item V:View S:Search I:Issue R:Return Q:QuitQ
----------------------------------------------------------------------
```

Hardware & Software Requirement

Hardware Requirement

PC/Laptop/MacBook with
Intel core/i3/i5/i7 or any equivalent
With at least 2 GB RAM
10 MB free space on Hard Disk
LCD/LED

Operating System & Compiler

MS Windows/Ubuntu/MacOS

Python IDLE 3.x
OR
colab.research.google.com (gmail account)

and/or
MySQL 8.x