Computer Science Project File

Library Management System

2021-2022

Asian International Private School

Ruwais, Abu Dhabi, UAE

NAME: Rakim Middya

GRADE: XII

REGISTER NO: 27615221

GUIDED BY: Ms. Reshma Premarajan

ASIAN INTERNATIONAL PRIVATE SCHOOL-RUWAIS

CERTIFICATE

This is to certify that Master **Rakim Middya of Grade 12**, Registration no: 27615221 has carried out the project work in Library Management Program (Python and SQL connectivity) prescribed by the Central Board of Secondary Education, New Delhi during the academic year 2021-22.

Teacher –in-charge : …….

Date: ……..

Internal Examiner: ……..

External Examiner: ……..

Principal: **School Seal:**

Table of Contents

Acknowledgement

In the accomplishment of this project, many people have bestowed upon their blessings and their heart pledged support.

Primarily I thank God Almighty for being able to complete this project with success. Then I would like to thank the management, my Principal Mr. Anzar Abdul Salam and my Computer Science teacher Ms. Reshma Premarajan whose valuable guidance and support has helped me bring out this project. Their suggestions and instructions have served me towards the completion of this project.

I would also like to thank my parents and friends for encouraging and helping me during the various phases of this project. Finally, I would like to thank CBSE for giving me this opportunity to undertake this project.

Objective

The project aims to provide a system to digitize library management for schools while still retaining all the features that come from doing it on paper.

Abstract

Startscreen:
When the program first boots up, it welcomes the user and displays a menu contains the following options:

1. Login
2. Register
3. Admin Login
4. Exit

- Login
  Prompts the user to enter username and password. If it's incorrect then it gives the option to go back to the startscreen. If the username and password match the one in database then user is enters the library and is shown the menu

- Register
  Prompts the user to enter their name and password. It then makes the database entries and sends user to the login screen directly. If the username already exists in the database, then the user is shown the same and is sent back to the startscreen.

- Menu:
  Upon logging in successfully the user is shown the following menu
  1. Book List
  2. Issue a book
  3. Return a book
  4. Go back to the login screen
  5. Exit

- Book List
  It displays all the books available in the library

- Issue a book
  Crosschecks username to see if the user had issued a book at an earlier date. If not, it prompts the user to enter the name of the book and makes an entry in the database with date, book name and name of the user. If the user has already issued a book, then they aren't allowed to issue another before returning the book

- Returning a book
  Checks the username of the person logged in the database. If the names match then they're prompted to return the book. Else they're sent back to the menu

- Admin Login

  Prompts the admin for username and password and checks it with the data stored on admin.txt. If the admin.txt doesn't exist then the admin is sent to the admin registration menu.

Upon logging in successfully the admin is shown the following menu:

1. See pending books
2. Lend a book
3. Add a new book to the database
4. Edit info of Existing Books
5. Return a book
6. Go back to the Startscreen
7. Show books
8. Damaged Books
9. Delete Books
10. Exit

- See pending books
  Shows a list of all the students that have issued a book along with the date and book name

- Lend a book
  Prompts the admin to enter the name of the student and name of the book and makes the entries in the database

- Add a new book to the database
  Prompts the admin to the name of the book and makes the necessary entries in the database

- Edit info of Existing Books
  Displays a list of books. Prompts the admin to enter the book-id and then gives the option to change the book name.
  After that the admin is prompted for confirmation. If the admin chooses yes, then the necessary changes are made to the database.

- Returning a book
  Prompts the admin to enter the name of the student. If the name exists in the database, program shows the book name and gives the option to return the book. If not then the admin is sent back to the menu

- Delete Books
  Displays all the books available in the library and prompts the user to enter the book name. If the book exists, it gets removed from the database and the admin is sent back to admin menu.

- Damaged Books
  Upon choosing this option the user is shown the following menu:
  1. Show damaged books
  2. Add damaged books
  3. Remove damaged books
  4. Return to menu

- Show damaged books: Shows damaged books

- Add damaged books: Prompts the user to enter book name and adds it to database

- Remove damaged books: Prompts to enter book name and removes it from database.

- Return to the menu: Upon choosing this option user is sent back to the menu.

Packages Used

mysql-connector-python

tabulate

pandas

datetime

csv

Files Generated

Admin.txt: Contains the login details of the admin.

login.csv : Contains the login details of all registered users.

damaged_books.csv: Contains details of all the damaged books in the library.

Methods Used

SQL functions:

Register():
Makes an entry into a sql table student_details with the user entered username and password.

Login():
Takes username and password as an input and then cross checks with the sql table. If the username/password are not a match then the user is given two options: either to register or try logging in again.

ShowBooks():
Displays all the books available in the library at that moment. Also had the option to show book that have been lended out as well

lendingdetails():
Takes username as a parameter to see if the user had issued a book at an earlier date. If not, it requests the book name and makes an entry in the database with date, book name and name of the user.
If the user has already issued a book, then they aren't allowed to issue another before returning the book

return_book():
Takes username as a parameter to see if the user had issued a book at an earlier date. If so, it'll check the number of days it was lent. If the number of days lent is over 7days then they're issued a penalty and only the admin can return their books now after collecting the penalty.
Penalty is 2 AED for every day over 7 days.
Otherwise the user is sent back to the menu screen.

deletebooks():
Takes book name as input and then deletes it from both book and booksavailable table. If there was nothing changed in the table then the output is "<bookname> wasn't found in the database".

DFH methods:

adminlogin()
Takes username and password as an input and cross checks with the username stored in admin.txt
If admin.txt doesn't exist then it calls for another function, adminregistration().

adminregistration()
Takes username and password as input and writes on to admin.txt.
It can only be accessed when the admin.txt file doesn't exist.

displaycsvtable():
Takes path to a csv file as an input and then outputs it as a table using Tabulate.

addbook():
takes book name as an input, writes it on to "damaged_books.csv" and deletes it from booksavailable table in the sql database.

deletebooksfromcsv():
Takes book name as an input, deletes it from the csv file and inserts the data into sql table.

Python Methods:

startscreen() : Prints welcome screen. Gives choice to either login, register or admin login

menu(): Prints menu after student logins successfully. Gives choice to either issue a book, return a book, see the books available or to just exit after

Source Code

```
#Library Management System
#Rakim Middya 12-B

import mysql.connector as mycon
from tabulate import tabulate
import datetime as dt
import os
import csv
import pandas as pd



mydb = mycon.connect(host = "localhost",
port = 3306,
user="root",
password = "root",
database = "test")
mycursor = mydb.cursor(buffered=True)
mycursor.execute('use test;')


def startscreen():
    print(' ============================================')
    print("|              School Library                |")
    print(' ============================================\n')

    print('Menu:\n─────\n')
    print('1. Login')
    print('2. Register')
    print('3. Admin Login')
    print('4. Exit\n')
    while True:
        choice = input("enter your choice(number): ")
        if choice == '1':
            login()
        elif choice == '2':
            register()
        elif choice == '3':
            adminlogin()
        elif choice == '4':
            print('Thank you for using the School Library!')
            exit()
        else:
            print('Please choose a number from 1,2,3,4\n')

def login():
```

```
    print('\nLogin:\n------\n')
    global username
    username = input("Username: ")
    password = input("Password: ")
    mycursor.execute('use test;')
    mycursor.execute('select * from student_details;')
    t1 = dict(mycursor.fetchall())
    if password == t1.get(username):
        print(f'\nHello, {username}!')
        menu()
    else:
        choice = input('Either the username or the password is
wrong\nWould you like to continue to try logging in?(Y or N):
')
        if choice.lower() == 'y':
            login()
        elif choice.lower() == 'n':
            startscreen()

def register():
    print('\nRegistration   \n----------\n')
    while True:
        username = input('Enter desired Username: ')
        mycursor.execute('use test;')
        mycursor.execute('Select * from Student_Details;')
        if username in dict(mycursor.fetchall()):
            choice = input('Sorry but this username already
exists in the database.Please contact the librarian for furthur
help\nWould you like to go to login screen?(Y or N): ')
            if choice.lower() == 'y':
                login()
            elif choice.lower() == 'n':
                startscreen()

        password = input('Enter password: ')
        sql = "Insert into student_details (name,password)
Values (%s,%s)"
        values = username,password
        mycursor.execute(sql, values)
        mydb.commit()
        print('Account', username,'is created successfully!')
        input("Press enter to continue")
        login()

def adminlogin():
    if os.path.isfile('admin.txt'):
        print('\nAdmin Login\n----------\n')
        global username
        while True:
            username = input('enter username: ')
            password = input('enter password: ')
```

```
            with open('admin.txt','r') as f:
                data = eval(f.readline())
                if username in data:
                    if data[username] == password:
                        adminmenu()
                if username not in data:
                    print('Either username or password is
incorrect.\nWould you like to continue trying logging in?')
                    choice = input('Choice (y or n): ')
                    choice = choice[0]
                    while True:
                        if choice.lower() == 'n':
                            startscreen()
                        elif choice.lower() == 'y':
                            break
                        else:
                            print('Please either choose either
y(es) or n(o).')
    else:
        adminregistration()

def adminregistration():
    fad = len('Admin Registration')*'—'
    print('\nAdmin Registration\n%s\n'%fad)
    global username
    username = input('enter username: ')
    password = input('enter password: ')
    dict1 = {}
    f = open('admin.txt','w')
    dict1[username]=password
    f.write(str(dict1))
    f.close()
    adminlogin()

def menu():
    global username
    while True:
        choice = input('\nWelcome to the school library\n1.Book
List\n2.Issue a book\n3.Return a book\n4.Go back to the login
screen\n5.Exit\nChoose a number(1,2,3,4,5): ')
        if choice == '1':
            showbooks()
        elif choice == '2':
            lendingdetails(username)
        elif choice == '3':
            return_book(username)
        elif choice == '4':
            print("\n")
            startscreen()
        elif choice == '5':
            print('Thank you for using the school library!')
```

```
            exit()
        else:
            print('please input either 1 or 2 or 3 or 4 or 5')

def showbooks():
    choice = '0'
    while True:
        if choice == '0':
            choice = input('\nWould you like to see-\n1) Only
the available books for lending\nor\n2) All the books that our
library has to offer\nChoice(1 or 2): ')
        if choice == '1':
            mycursor.execute('select      book_name      from
booksavailable')

print(tabulate(mycursor.fetchall(),headers=["Name"],tablefmt="
grid"))
            break
        elif choice == '2':
            mycursor.execute('select * from book')
            print(tabulate(mycursor.fetchall(),headers=["Book
ID","Name"],tablefmt="grid"))
            break
        else:
            choice = input('Please choose either 1 or 2.')

    input('\npress enter to back to the menu\n')
    menu()

def lendingdetails(name = 'abc'):
    if name == 'abc':
        name = input('Enter the name of the student: ')
    mycursor.execute("select  *  from  lending  where  name
='%s';"%name)

    if len(mycursor.fetchall())!= 0:
        print('\nYou have already issued a book.')
        print('Please  return  it  before  trying  to  issue
another.')
        input('Returning  to  the  menu...(press  enter  to
continue)')
        menu()
    print('\nBooks Available\n---------------')
    mycursor.execute('Select book_name from booksavailable;')

print(tabulate(mycursor.fetchall(),headers=["Name"],tablefmt="
grid"))
    mycursor.execute("Select  *  from  lending  where  name
='%s';"%name)
    book_name = input('Enter name of the book: ')
```

```
    mycursor.execute("Select * from booksavailable where
book_name ='%s';"%book_name)
    if(len(mycursor.fetchall()) != 0):
        date                                                =
dt.datetime.now().strftime('%d'+'/'+'%m'+'/'+'%Y')
        values = date,name,book_name
        sql = "Insert into lending (date,name,book_name) Values
(%s,%s,%s)"
        mycursor.execute(sql, values)
        print(name,'has taken',book_name,'on',date)
        mycursor.execute("Delete from booksavailable where
book_name ='%s'"%book_name)
        mydb.commit()
        input('\npress enter to back to the menu\n')
    else:
        mycursor.execute("Select * from book where book_name
='%s'"%book_name)
        if(len(mycursor.fetchall()) != 0):
            print("I am sorry but the book requested is
currently lent out to someone. Please try at a later date.")
            input("Press enter to continue")
        else:
            print(book_name,"doesn't exist in our library.
Please ask the Librarian if you would like it to be included.")
            input("Press enter to continue")
    mydb.commit()
    menu()

def return_book(name = 'abc'):
    print('\nBook return\n-----------\n')
    if name == 'abc':
        name = input('enter your name/name of the student: ')

    mycursor.execute("Select book_name from lending where
name='%s';"%name)
    data = mycursor.fetchall()
    if(len(data)==0):
        print("You haven't taken any book yet. Please issue a
book before choosing three")
        input('Returning to menu....')
        menu()
    else:
        book_name = data[0][0]
        days = dayslent(name)
        if days < 7:
            choice    =    input('Would    you    like    to
return %s\nChoice(y(es) or n(o)): '%book_name)
            if choice[0].lower() == 'y':
                sql  =  "DELETE  FROM  lending  WHERE  name  =
'%s'" %name
                mycursor.execute(sql)
```

```
                sql          =          "Insert          into
booksavailable(book_id,book_name) Values (%s,%s)"
                mycursor.execute("select * from book;")
                books = dict(mycursor.fetchall())
                id_num = 0
                for i in books:
                    if books[i] == book_name[1:-1]:
                        id_num = i
                val = (id_num,book_name)
                mycursor.execute(sql,val)
                mydb.commit()
                print('%s     has     returned     book     %s
sucessfully'%(name,book_name))
                input('Returning to the menu...(press enter to
continue)')
                mydb.commit()
                menu()
        else:
            print('You have incurred a fine of %s'%str((days-
7)*2)+f" AED (2*({days}-7)) for not returning the book on
time.\nYou are not allowed to take another book until you pay
it. Please contact the librarian for furthur instruction.\n")
            input('Returning to the menu...(press enter to
continue)')
            menu()




def adminmenu(foo=None):
    def menu():
            print('\n1. See pending books')
            print('2. Lend a book')
            print('3. Add a new book to the database')
            print('4. Edit info of Existing Books')
            print('5. Return a book')
            print('6. Go back to the Startscreen')
            print('7. Show books')
            print('8. Damaged Books')
            print('9. Delete Books')
            print('10. Exit\n')
            choice = input("enter your choice: ")
            if choice == '1':
                see_lent()
            elif choice == '2':
                adminlendingdetails()
            elif choice == '3':
                enterbookinfo()
            elif choice == '4':
                updatebookinfo()
```

```
            elif choice == '5':
                adminreturn_book()
            elif choice == '6':
                startscreen()
            elif choice == '7':
                adminshowbooks()
            elif choice == '8':
                damagedbooks()
            elif choice == '9':
                deletebooks()
            elif choice == '10':
                print('Thank you for using the school library!')
                exit()

            else:
                print("\nPlease choose a number from the
following list- [1,2,3,4,5,6,7,8]: ")
                adminmenu('fdasfds')
    if foo == None:
        global username
        print('\nAdmin Menu\n——————————\n')
        print("Welcome",username)
        print('Choose a number')
        menu()
    else:
        print('\nAdmin Menu\n——————————\n')
        print('Choose a number')
        menu()

def see_lent():
    mycursor.execute("select * from lending;")
    data = mycursor.fetchall()
    if len(data) != 0:
        print(tabulate(data,headers=["Date","Name","Book
Name"],tablefmt="grid"))
        input('Press enter to continue')
    else:
        print("There are no books pending!\n")
        input('Press enter to continue')
    adminmenu('x')

def enterbookinfo():
    print('\nBook Info\n')
    while True:
        book_name = input('enter name of the book: ')
        mycursor.execute('Select * from book;')
        if book_name in dict(mycursor.fetchall()):
            print('This book already exists in the database.')
            def choice():
                choice = input("\nWould you like to try a
different book?\nChoice(y,n)= ")
```

```
                if choice[0].lower() == 'y':
                    enterbookinfo()
                elif choice[0].lower() == 'n':
                    adminmenu()
                else:
                    print('please either y(es) or n(o): ')
                    choice()
        sql    =    "Insert    into    book(book_name)    Values
('%s');"%book_name
        mycursor.execute(sql)
        mydb.commit()

        sql  =  "Insert  into  booksavailable(book_name)  Values
('%s');"%book_name
        values = book_name
        mycursor.execute(sql, values)
        mydb.commit()

        print('The entry for',book_name,'has been successfully
created.\n')
        choice  =  input('Would  you  like  to\n1.  Add  another
book\n2. Go back to the menu\nChoose an option(1 or 2): ')
        if choice == '1':
            continue
        elif choice == '2':
            adminmenu()

def updatebookinfo():
    mycursor.execute('select * from book')
    print(tabulate(mycursor.fetchall(),headers=["Book
ID","Name"],tablefmt="grid"))
    id = input('enter book ID: ')
    mycursor.execute('select book_name from book where book_id
=%s' %id)
    if len(mycursor.fetchall()) == 0:
        print('The book id entered does not exist')
        choice = input("Would you like to retype the id?\nChoice
(Y(es) or N(o)): ")
        if choice.lower()[0] == 'n':
            adminmenu()

    mycursor.execute('select book_name from book where book_id
=%s' %id)
    book_name = mycursor.fetchall()[0][0]
    newname = input('enter new name: ')
    val = (newname,id)
    query  =  "Update  book  set  book_name='%s'  where  book_id
=%s" %val+';'
    mycursor.execute(query)
    query  =  "Update  booksavailable  set  book_name='%s'  where
book_id =%s" %val+';'
```

```
    mycursor.execute(query)
    l1 = [["Old Name","New Name"],[book_name,newname]]
    print(tabulate(l1,headers="firstrow",tablefmt="grid"))
    choice = input('Would you like to go forward with the name
change shown above?\nChoice(yes or no): ')
    while True:
        if choice[0].lower() == 'y':
            mydb.commit()
            print('%s    has    been    successfully    changed
to %s'%(book_name,newname))
            input("Going  back  to  the  menu...(press  enter  to
continue)")
            adminmenu()
        elif choice[0].lower() == 'n':
            option    =    input('Do    you    want    to    try
again?\nChoice(yes or no): ')

            while True:
                if option[0].lower == 'y':
                    updatebookinfo()

                elif option[0].lower == 'n':
                    input("Going  back  to  the  menu...(press
enter to continue)")
                    adminmenu()
                else:
                    option = input('please type either yes or
no: ')

        else:
            choice = input('please type either yes or no')

def adminlendingdetails():
    name = input('Enter the name of the student: ')
    mycursor.execute("select   *   from   lending   where   name
='%s';"%name)
    if len(mycursor.fetchall())!= 0:
        print('\n%s has already issued a book.'%name)
        print('Please get him or her to return it.')
        input('Returning   to   the   menu...(press   enter   to
continue)')
        adminmenu()
    print('Books Available\n---------------')
    mycursor.execute('Select book_name from booksavailable;')

print(tabulate(mycursor.fetchall(),headers=["Name"],tablefmt="
grid"))
    mycursor.execute("Select   *   from   lending   where   name
='%s';"%name)
    book_name = input('Enter name of the book: ')
```

```
    mycursor.execute("Select   *   from   booksavailable   where
book_name ='%s';"%book_name)
    if(len(mycursor.fetchall()) != 0):
        date                                                    =
dt.datetime.now().strftime('%d'+'/'+'%m'+'/'+'%Y')
        values = date,name,book_name
        sql = "Insert into lending (date,name,book_name) Values
(%s,%s,%s)"
        mycursor.execute(sql, values)
        print(name,'has taken',book_name,'on',date)
        mycursor.execute("Delete   from   booksavailable   where
book_name ='%s'"%book_name)
        mydb.commit()
        input('\npress enter to back to the menu\n')
    else:
        mycursor.execute("Select  *  from  book  where  book_name
='%s'"%book_name)
        if(len(mycursor.fetchall()) != 0):
            print("I  am  sorry  but  the  book  requested  is
currently lent out to someone. Please try at a later date.")
            input("Press enter to continue")
        else:
            print(book_name,"doesn't exist in our library.")
            input("Press enter to continue")
    mydb.commit()
    adminmenu()

def adminshowbooks(choice = '0'):
    while True:
        if choice == '0':
            choice = input('\nWould you like to see-\n1) Only
the available books for lending\nor\n2) All the books that our
library has to offer\nChoice(1 or 2): ')
        if choice == '1':
            mycursor.execute('select      book_name      from
booksavailable')

print(tabulate(mycursor.fetchall(),headers=["Name"],tablefmt="
grid"))
            break
        elif choice == '2':
            mycursor.execute('select * from book')
            print(tabulate(mycursor.fetchall(),headers=["Book
ID","Name"],tablefmt="grid"))
            break
        else:
            print('Please choose either 1 or 2.')

    input('\npress enter to back to the menu\n')
    adminmenu()
```

```
def adminreturn_book(name = 'abc'):
    print('Book return\n----------\n')
    if name == 'abc':
        name = input('enter the name of the student: ')
    mycursor.execute("Select  book_name  from  lending  where
name='%s';"%name)
    data = mycursor.fetchall()
    if(len(data)==0):
        print("%s hasn't taken any book yet."%name)
        input('Returning to menu....')
        adminmenu()
    else:
        book_name = data[0][0]
        days = dayslent(name)
        if days < 7:
            choice  =  input('Is  %s  returning  %s?\nChoice
[Y(es)/n(o)]: '%(name,book_name))
            if choice[0].lower() == 'y':
                sql  =  "DELETE  FROM  lending  WHERE  name  =
'%s'" %name
                mycursor.execute(sql)
                sql          =          "Insert          into
booksavailable(book_id,book_name) Values (%s,%s)"
                mycursor.execute("select * from book;")
                books = dict(mycursor.fetchall())
                id_num = 0
                for i in books:
                    if books[i] == book_name:
                        id_num = i
                val = (id_num,book_name)
                mycursor.execute(sql,val)
                mydb.commit()
                print('%s has returned book %s\nAnd the database
has been updated to reflect that'%(name,book_name))
                input('Returning to the menu...(press enter to
continue)')
        else:
            choice  =  input('Has  %s  paid  the  fine  of
'%name+str((days-7)*2)+' AED?\nChoice [Y(es)/n(o)]:')
            if choice[0].lower() == 'y':
                sql  =  "DELETE  FROM  lending  WHERE  name  =
'%s'" %name
                mycursor.execute(sql)
                sql          =          "Insert          into
booksavailable(book_id,book_name) Values (%s,%s)"
                mycursor.execute("select * from book;")
                books = dict(mycursor.fetchall())
                id_num = 0
                for i in books:
                    if books[i] == book_name:
```

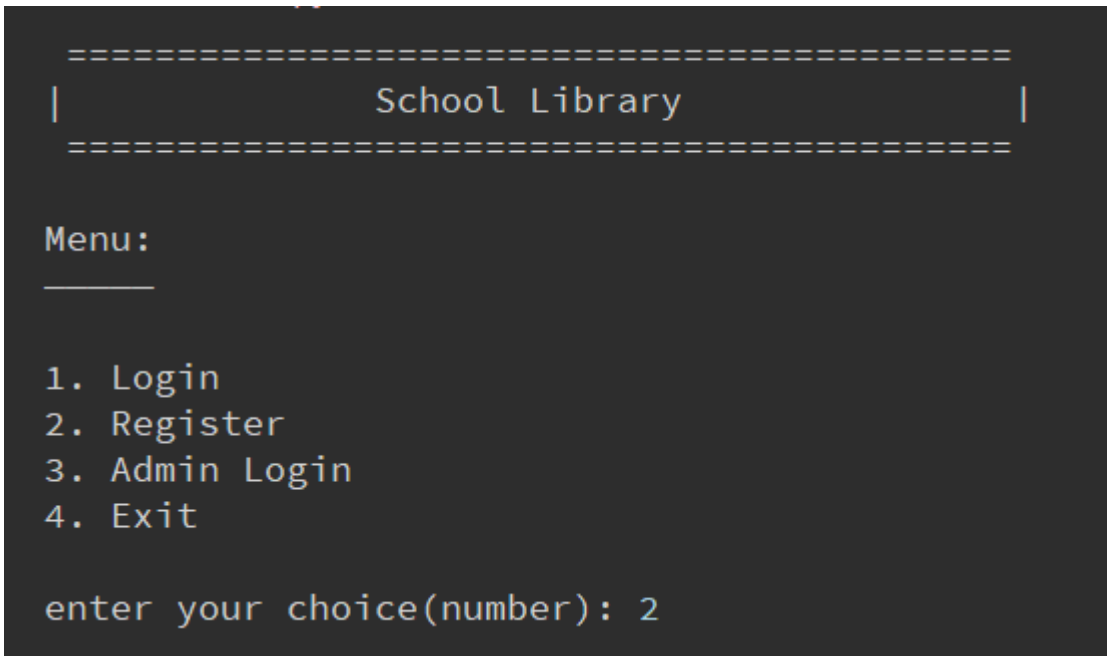
```
                        id_num = i
                val = (id_num,book_name)
                mycursor.execute(sql,val)
                mydb.commit()
                print('%s              has              returned
book %s'%(name,book_name))
                input('Returning to the menu...(press enter to
continue)')
    mydb.commit()
    adminmenu()


#new stuff

def dayslent(name):
    mycursor.execute('select  date  from  lending  where  name  =
"%s"'%name)
    lentdate = mycursor.fetchall()[0][0]
    date                                                    =
dt.datetime.strptime(lentdate,'%d'+'/'+'%m'+'/'+'%Y')
    difference   =   dt.datetime.now()-date   #calculates   the
difference
    return int(difference.days)

def damagedbooks():
    print('\nDamaged Books\n----------------\n')
    print('1. Show damaged books')
    print('2. Add damaged books')
    print('3. Remove damaged books')
    print('4. Return to menu\n')
    choice = input('Enter choice: ')
    if choice == '1':
        displaycsvtable()
    if choice == '2':
        addbook()
    if choice == '3':
        deletebooksfromcsv()
    if choice == '4':
        adminmenu()

def displaycsvtable(x=None):
    print('\nDamaged Books\n-------------')
    filename = 'damaged_books.csv'
    data = []
    with open(filename,'r',newline='') as damaged_books:
        csvreader = csv.reader(damaged_books)
        for i in csvreader:
            data.append(i)
    print(tabulate(data,headers="firstrow",tablefmt='grid'))
    if x == None:
        input('\npress enter to back to the menu\n')
```

```
        damagedbooks()
    else:
        pass


def addbook():
    filename = 'damaged_books.csv'
    with open(filename,'a',newline='') as damaged_books:
        csvwriter = csv.writer(damaged_books)
        data = pd.read_csv(filename).to_numpy()
        bookname = input("Enter bookname: ")
        mycursor.execute("Delete  from  booksavailable  where
book_name ='%s'"%bookname)
        mydb.commit()
        bookid =  data[-1][0]+1
        csvwriter.writerow([bookid,bookname])
    input('\npress enter to back to the menu\n')
    damagedbooks()




def deletebooksfromcsv():
    filename = 'damaged_books.csv'
    displaycsvtable(1)
    data = []
    name = input("Enter the book name: ")

    with open(filename,'r',newline='') as damaged_books:
        csvreader = csv.reader(damaged_books)
        for i in csvreader:
            data.append(i)
    for i in range(len(data)):
        if data[i][1] == name:
            del data[i]
            break

    with open(filename,'w',newline='') as damaged_books:
        csvwriter = csv.writer(damaged_books)
        header = ["Serial No.","Book Name"]
        csvwriter.writerow(header)
        for i in range(1,len(data)):
            bookname = data[i][1]
            bookid =  i
            csvwriter.writerow([bookid,bookname])
    mycursor.execute('Insert  into  booksavailable(book_name)
Values("%s");'%name)
    mydb.commit()


    displaycsvtable()
```

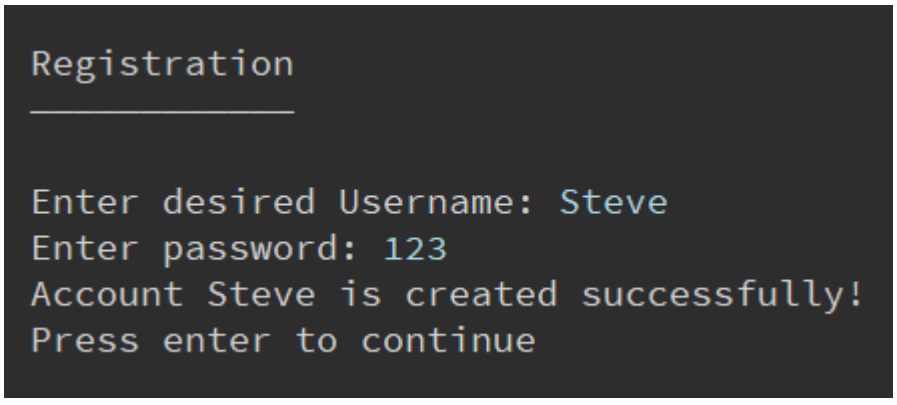
```
    input('\npress enter to back to the menu\n')
    damagedbooks()


def deletebooks():
    mycursor.execute('select * from book')
    print(tabulate(mycursor.fetchall(),headers=["Book
ID","Name"],tablefmt="grid"))
    name = input("Enter book-name: ")
    mycursor.execute(f"Delete   from   booksavailable   where
book_name = '{name}'")
    mycursor.execute(f"Delete  from  book  where  book_name  =
'{name}'")
    mydb.commit()
    if mycursor.rowcount != 0:
        print(f"{name} has been deleted successfully.")
        input("\npress enter to continue")

        adminmenu()
    else:
        print(f"{name} was not found in the database.")
        input("\npress enter to continue")
        adminmenu()

startscreen()
exit()
```

Output Screens

1. Welcome Screen



2. Registration Screen



If the username already exists:

```
Registration
____________

Enter desired Username: Steve
Sorry but this username already exists in the database.Please contact the librarian for furthur help
```

3. Login Screen

Correct Username+Password

```
Login:
______

Username: Steve
Password: 123

Hello, Steve!
```

Login screen: Incorrect Username+Password

```
Login:
______

Username: Steve
Password: 235
Either the username or the password is wrong
Would you like to continue to try logging in?(Y or N): Y
```

4. Main menu

```
Welcome to the school library
1.Book List
2.Issue a book
3.Return a book
4.Go back to the login screen
5.Exit
Choose a number(1,2,3,4,5):
```

5. Book List:

```
Would you like to see-
1) Only the available books for lending
or
2) All the books that our library has to offer
Choice(1 or 2): 1
+--------------------+
| Name               |
+====================+
| The Hobbit         |
+--------------------+
| The Wheel of Time  |
+--------------------+
| No Longer Human    |
+--------------------+
| Born a Crime       |
+--------------------+
| Blank Canvas       |
+--------------------+
```

```
Would you like to see-
1) Only the available books for lending
or
2) All the books that our library has to offer
Choice(1 or 2): 2
+-----------+--------------------+
|   Book ID | Name               |
+===========+====================+
|         1 | The Hobbit         |
+-----------+--------------------+
|         2 | The Wheel of Time  |
+-----------+--------------------+
|         3 | No Longer Human    |
+-----------+--------------------+
|         4 | Born a Crime       |
+-----------+--------------------+

press enter to back to the menu
```

6. Issuing a book:

```
Choose a number(1,2,3,4,5): 2
Books Available
---------------
+--------------------+
| Name               |
+====================+
| The Hobbit         |
+--------------------+
| The Wheel of Time  |
+--------------------+
| No Longer Human    |
+--------------------+
| Born a Crime       |
+--------------------+
| Blank Canvas       |
+--------------------+
Enter name of the book: The Hobbit
Steve has taken The Hobbit on 03/02/2022

press enter to back to the menu
```

7. Returning a book:

```
Book return
-----------

Would you like to return The Hobbit
Choice(y(es) or n(o)): yes
Steve has returned book The Hobbit sucessfully
```

8. Going back to the welcome screen:

```
Welcome to the school library
1.Book List
2.Issue a book
3.Return a book
4.Go back to the login screen
5.Exit
Choose a number(1,2,3,4,5): 4


 ========================================
|             School Library             |
 ========================================

Menu:
_____

1. Login
2. Register
3. Admin Login
4. Exit
```

ADMIN MENU

```
Admin Menu
__________

Welcome Admin
Choose a number

1. See pending books
2. Lend a book
3. Add a new book to the database
4. Edit info of Existing Books
5. Return a book
6. Go back to the Startscreen
7. Show books
8. Damaged Books
9. Delete Books
10. Exit
```

1. See pending books:

```
enter your choice: 1
+------------+--------+-------------------+
| Date       | Name   | Book Name         |
+============+========+===================+
| 04/02/2022 | Jacob  | The Wheel of Time |
+------------+--------+-------------------+
Press enter to continue
```

2. Lending a book:

```
enter your choice: 2
Enter the name of the student: Jacob
Books Available
--------------
+------------------+
| Name             |
+==================+
| The Wheel of Time |
+------------------+
| No Longer Human   |
+------------------+
| Born a Crime      |
+------------------+
| Blank Canvas      |
+------------------+
| The Hobbit        |
+------------------+
Enter name of the book: The Wheel of Time
Jacob has taken The Wheel of Time on 04/02/2022

press enter to back to the menu
```

```
enter your choice: 3

Book Info

enter name of the book: Golden Kamuy
The entry for Golden Kamuy has been successfully created.
```

3. Add a new book to the database

```
enter your choice: 3

Book Info

enter name of the book: The Eye of the World
The entry for The Eye of the World has been successfully created.
```

4. Editing the book name:

```
enter your choice: 4
+-----------+------------------+
|   Book ID | Name             |
+===========+==================+
|         1 | The Hobbit       |
+-----------+------------------+
|         2 | The Wheel of Time |
+-----------+------------------+
|         3 | No Longer Human  |
+-----------+------------------+
|         4 | Born a Crime     |
+-----------+------------------+
|         6 | Golden Kamuy     |
+-----------+------------------+
enter book ID: 1
enter new name: The Lord of the Rings- Book 1
+------------+-------------------------------+
| Old Name   | New Name                      |
+============+===============================+
| The Hobbit | The Lord of the Rings- Book 1 |
+------------+-------------------------------+
Would you like to go forward with the name change shown above?
Choice(yes or no): yes
The Hobbit has been successfully changed to The Lord of the Rings- Book 1
Going back to the menu...(press enter to continue)
```

5. Returning book as an Admin

```
enter your choice: 5
Book return
-----------

enter the name of the student: Jacob
Is Jacob returning The Wheel of Time?
Choice [Y(es)/n(o)]: yes
Jacob has returned book The Wheel of Time
```

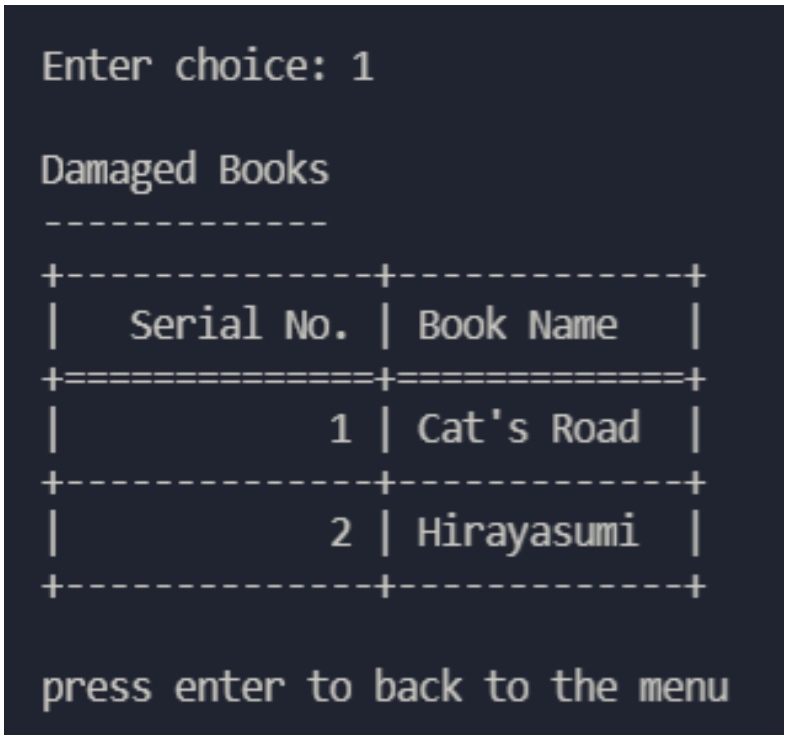
6. Deleting books from the database

```
enter your choice: 9
+-----------+------------------------------+
|   Book ID | Name                         |
+===========+==============================+
|         1 | The Lord of the Rings- Book 1 |
+-----------+------------------------------+
|         2 | The Wheel of Time            |
+-----------+------------------------------+
|         3 | No Longer Human              |
+-----------+------------------------------+
|         4 | Born a Crime                 |
+-----------+------------------------------+
|         6 | Golden Kamuy                 |
+-----------+------------------------------+
Enter book-name: The Lord of the Rings- Book 1
The Lord of the Rings- Book 1 has been deleted successfully.

press enter to continue
```

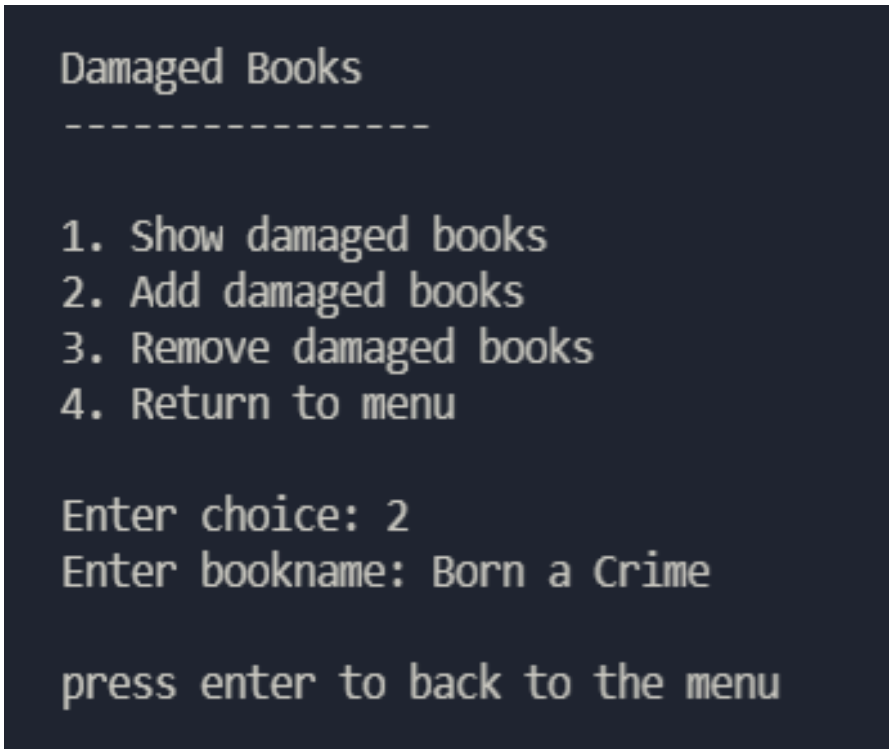
Damaged Books:

```
enter your choice: 8

Damaged Books
---------------

1. Show damaged books
2. Add damaged books
3. Remove damaged books
4. Return to menu
```

1. Showing the damaged books



2. Adding damaged books

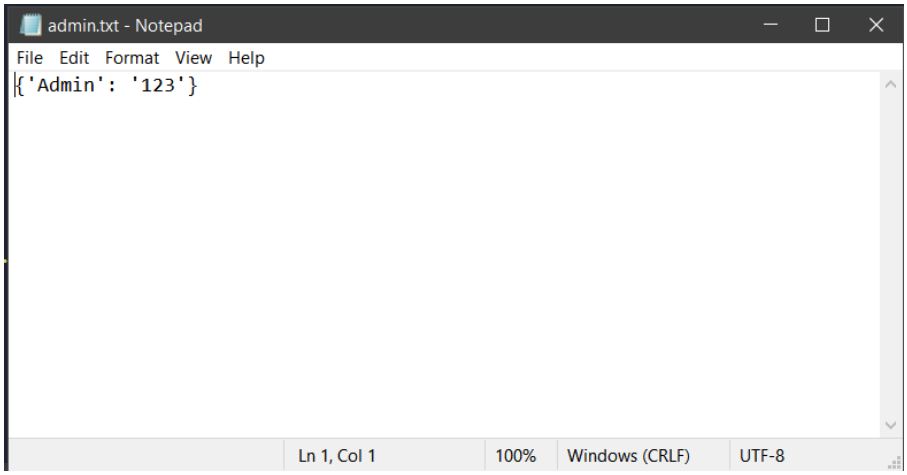


```
Damaged Books
---------------

1. Show damaged books
2. Add damaged books
3. Remove damaged books
4. Return to menu

Enter choice: 1

Damaged Books
-------------
+--------------+--------------+
|   Serial No. | Book Name    |
+==============+==============+
|            1 | Cat's Road   |
+--------------+--------------+
|            2 | Hirayasumi   |
+--------------+--------------+
|            3 | Born a Crime |
+--------------+--------------+

press enter to back to the menu
```

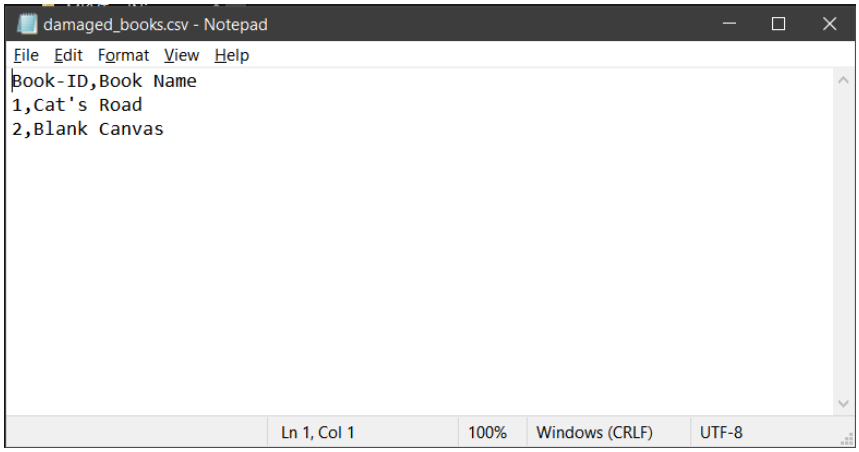
3. Deleting damaged books:

```
Damaged Books
---------------

1. Show damaged books
2. Add damaged books
3. Remove damaged books
4. Return to menu

Enter choice: 3

Damaged Books
-------------
+--------------+--------------+
|   Serial No. | Book Name    |
+==============+==============+
|            1 | Cat's Road   |
+--------------+--------------+
|            2 | Hirayasumi   |
+--------------+--------------+
|            3 | Born a Crime |
+--------------+--------------+
Enter the book name: Hirayasumi

Damaged Books
-------------
+--------------+--------------+
|   Serial No. | Book Name    |
+==============+==============+
|            1 | Cat's Road   |
+--------------+--------------+
|            2 | Born a Crime |
+--------------+--------------+

press enter to back to the menu
```

Txt files:



CSV file:



SQL tables:

student_details: contains username and password):

```
mysql> select * from student_details;
+-------+----------+
| name  | password |
+-------+----------+
| Steve | 123      |
+-------+----------+
1 row in set (0.00 sec)
```

book: contains entire list of books that the library has

```
mysql> select * from book;
+---------+-------------------+
| book_id | book_name         |
+---------+-------------------+
|       1 | The Hobbit        |
|       2 | The Wheel of Time |
|       3 | No Longer Human   |
|       4 | Born a Crime      |
+---------+-------------------+
```

booksavailable: contains only those books available for being issued

```
mysql> select * from booksavailable;
+---------+-------------------+
| book_id | book_name         |
+---------+-------------------+
|       2 | The Wheel of Time |
|       3 | No Longer Human   |
|       4 | Born a Crime      |
+---------+-------------------+
```

lending: contains list of books lent out along with their name and date of issuing

```
mysql> select * from lending;
+------------+-------+------------+
| date       | name  | book_name  |
+------------+-------+------------+
| 09/02/2022 | Steve | The Hobbit |
+------------+-------+------------+
```

Limitations

- Login details are saved in plain text.
- All the databases including SQL can be accessed and modified externally if the user has sufficient permissions.
- No support for multiple students with the same name.
- The system inherently relies on trust. There's no check or verification for whether the student has returned the book or has he/she/they used his actual name while creating his account.

Requirements

❖ HARDWARE

- Processor: Intel Core i7-10510U CPU @ 1.80GHz
- RAM: 32GB
- Hard Disk: 512GB.
- Keyboard and mouse/touchpad

❖ SOFTWARE

- Operating system: Windows, macOS, Debian, Fedora
- Programming Language: Python
- Application used for coding: VSCode, Python IDLE, PyCharm, Anaconda

Bibliography

- https://www.google.com/
- https://www.geeksforgeeks.org/
- Computer Science with Python: SUMITA ARORA