System Description

For

Library Management System

Team Members:

1.Nourhan Ayman Abdel-Moati(team leader)

2.Heba Maher Al-Shahat

3.Nada Yasser Morsi

4.Nourhan Ayman Al-sayed

Introduction:

This system is designed to help students with the ability to easily access the books they need for their academic journey for free, with this system students can borrow books, each book has an id, the system enable admin to update and add new books to the collection of books. Each student borrow book has a specific period for borrowing book.

Scope:

This system aims to simplify the operation of borrowing and management of books, the admin is only the person can access the system, the student orders the book, the admin will go to issue book feature and write the information of book in list of borrowed books, each book has id, title, date that is borrowed, author, also enter the information of student: phone and name. When the student returns the book, the admin enters the information about that book and student name and updates the available book.

Features:

1.Admin is only person can access the information of this system which guarantees security
2.list of available books that students need in their academic journey for free.
3.Easily to borrow any book
4.Easily to return book
5. Ability for admin to update books, if new books become available

Tools:

1.softwatre programs: visual studio code, Spyder for coding

2.programming language: python

3.software program for database: SQLite

4.software program to collaboration team members with each other: GitHub

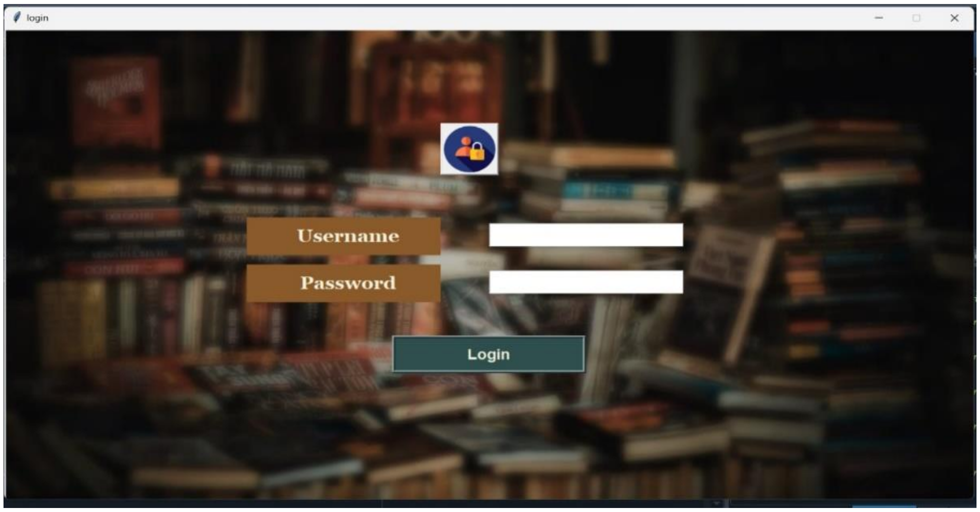
Used libraries:

- **tkinter**
- **from PIL (ImageTk,Image)**
- **importlib**

Interfaces:

Login interface

This login interface to the admin, contains username, password knownonly by him.
If the username and password are correct,the login button, will go to the home page, and select where to go.

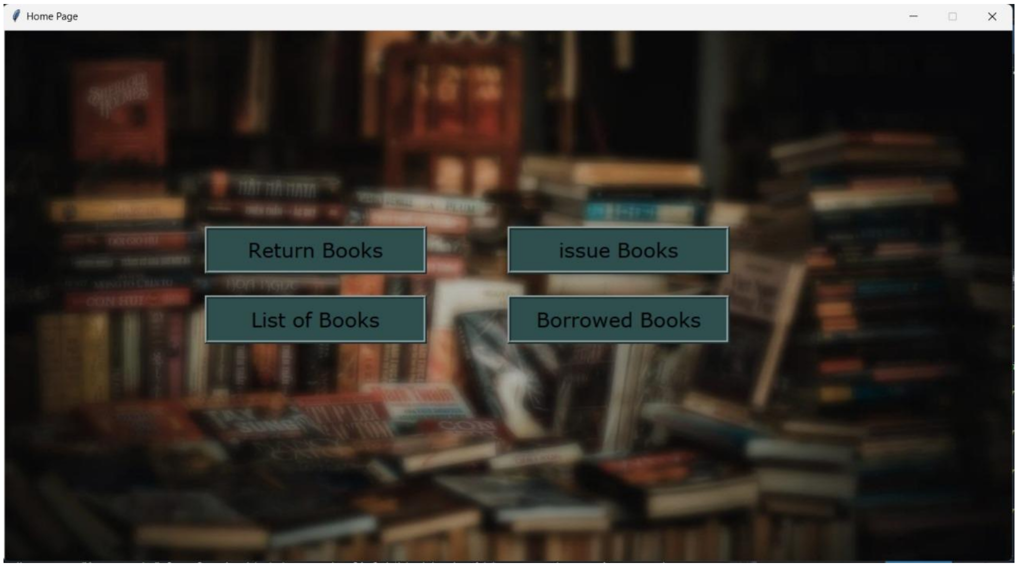


Home Page

Home page has four options:

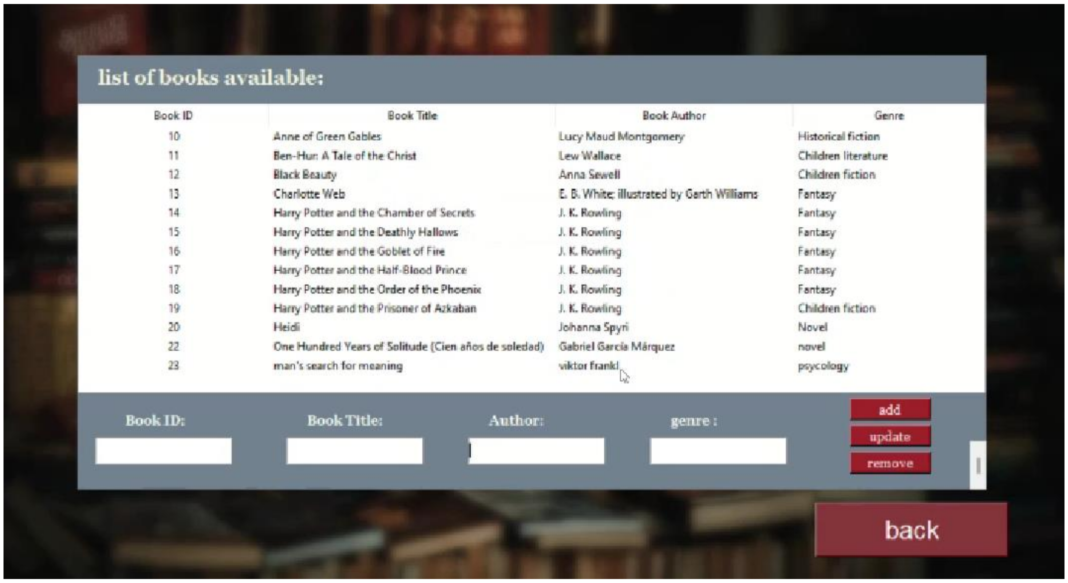
-Return book, issue book and borrowed books will go to the list of borrowed books interface where he can issue or return a book.

-list of books go to the interface of the available books in the library.



List of books

This frame show that available books, the information about each book , book_id, book_title, Book author, genre, author, and contain add button , to add a new book, update book to update an information of a book.



list of books available:

Book ID	Book Title	Book Author	Genre
10	Anne of Green Gables	Lucy Maud Montgomery	Historical fiction
11	Ben-Hur: A Tale of the Christ	Lew Wallace	Children literature
12	Black Beauty	Anna Sewell	Children fiction
13	Charlotte Web	E. B. White; illustrated by Garth Williams	Fantasy
14	Harry Potter and the Chamber of Secrets	J. K. Rowling	Fantasy
15	Harry Potter and the Deathly Hallows	J. K. Rowling	Fantasy
16	Harry Potter and the Goblet of Fire	J. K. Rowling	Fantasy
17	Harry Potter and the Half-Blood Prince	J. K. Rowling	Fantasy
18	Harry Potter and the Order of the Phoenix	J. K. Rowling	Fantasy
19	Harry Potter and the Prisoner of Azkaban	J. K. Rowling	Children fiction
20	Heidi	Johanna Spyri	Novel
22	One Hundred Years of Solitude (Cien años de soledad)	Gabriel García Márquez	novel
23	man's search for meaning	viktor frankl	psycology

Book ID:

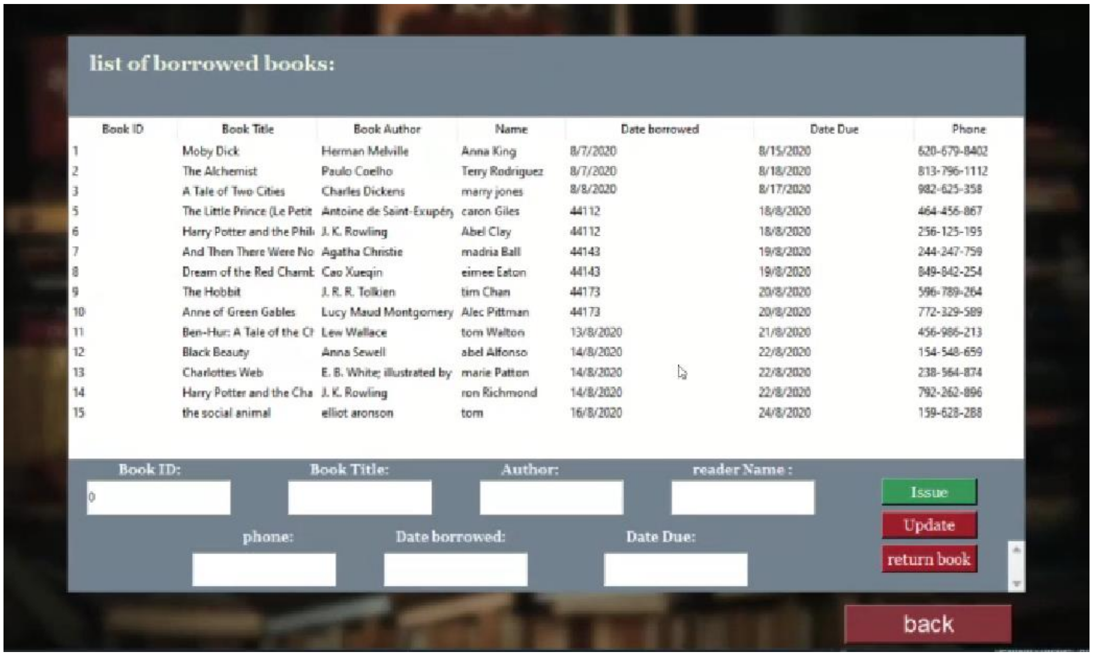
Book Title:

Author:

genre :

List of borrowed books

This frame shows the list of borrowed books ,option to issue a book ,update to update the list of borrowed book,or return the issued book.



list of borrowed books:

Book ID	Book Title	Book Author	Name	Date borrowed	Date Due	Phone
1	Moby Dick	Herman Melville	Anna King	8/7/2020	8/15/2020	620-679-8402
2	The Alchemist	Paulo Coelho	Terry Rodriguez	8/7/2020	8/18/2020	813-796-1112
3	A Tale of Two Cities	Charles Dickens	marry jones	8/8/2020	8/17/2020	982-625-358
5	The Little Prince (Le Petit	Antoine de Saint-Exupéry	caron Giles	44112	18/8/2020	464-456-867
6	Harry Potter and the Phil	J. K. Rowling	Abel Clay	44112	18/8/2020	256-125-195
7	And Then There Were No	Agatha Christie	madria Ball	44143	19/8/2020	244-247-759
8	Dream of the Red Chamb	Cao Xueqin	eimee Eaton	44143	19/8/2020	849-842-254
9	The Hobbit	J. R. R. Tolkien	tim Chan	44173	20/8/2020	596-789-264
10	Anne of Green Gables	Lucy Maud Montgomery	Alec Pittman	44173	20/8/2020	772-329-589
11	Ben-Hur: A Tale of the Cl	Lew Wallace	tom Walton	13/8/2020	21/8/2020	456-986-213
12	Black Beauty	Anna Sewell	abel Alfonso	14/8/2020	22/8/2020	154-548-659
13	Charlottes Web	E. B. White; illustrated by	marie Patton	14/8/2020	22/8/2020	238-564-874
14	Harry Potter and the Cha	J. K. Rowling	ron Richmond	14/8/2020	22/8/2020	792-262-896
15	the social animal	elliot aronson	tom	16/8/2020	24/8/2020	159-628-288

Book ID: 0

Book Title:

Author:

reader Name :

phone:

Date borrowed:

Date Due:

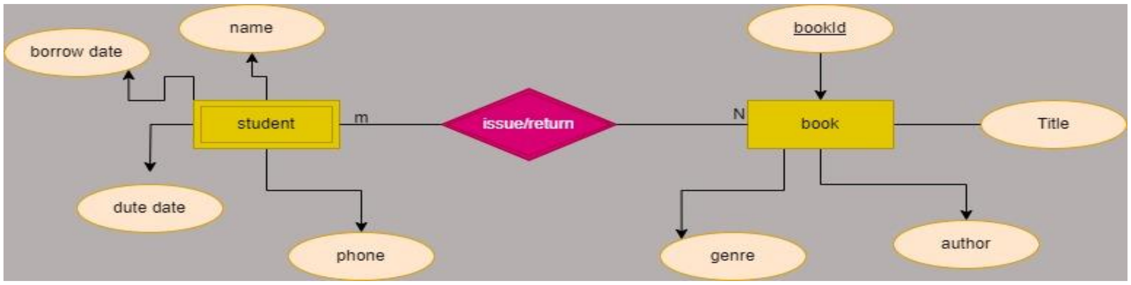
ERD:

Components of an erd:

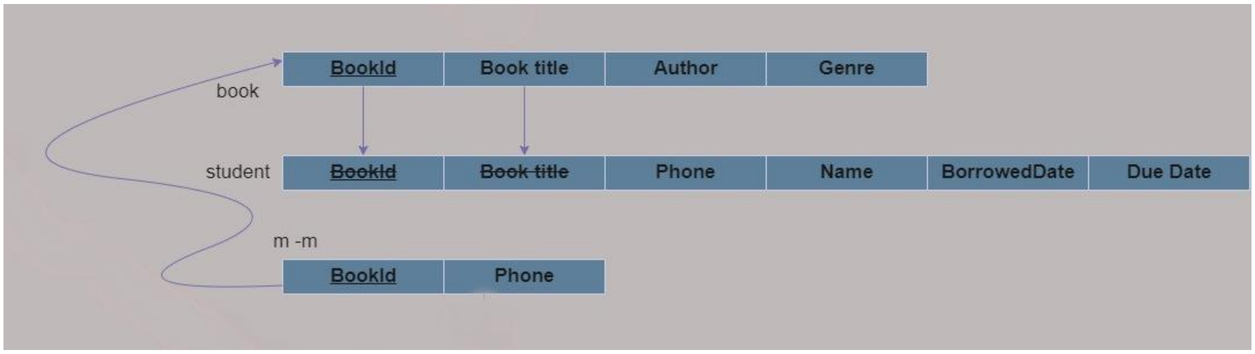
- Entities: student, book
- Attributes:1. Student(weak entity) has four attributes 1. Name 2. Borrow date 3. Dute date 4. Phone

2.Book (strong entity)has four attributes 1. Book_id(primary key). 2.author 3.genre 4.title

- Relationships: student can borrow or return multiple books, or book can be borrowed or returned by many student



Mapping

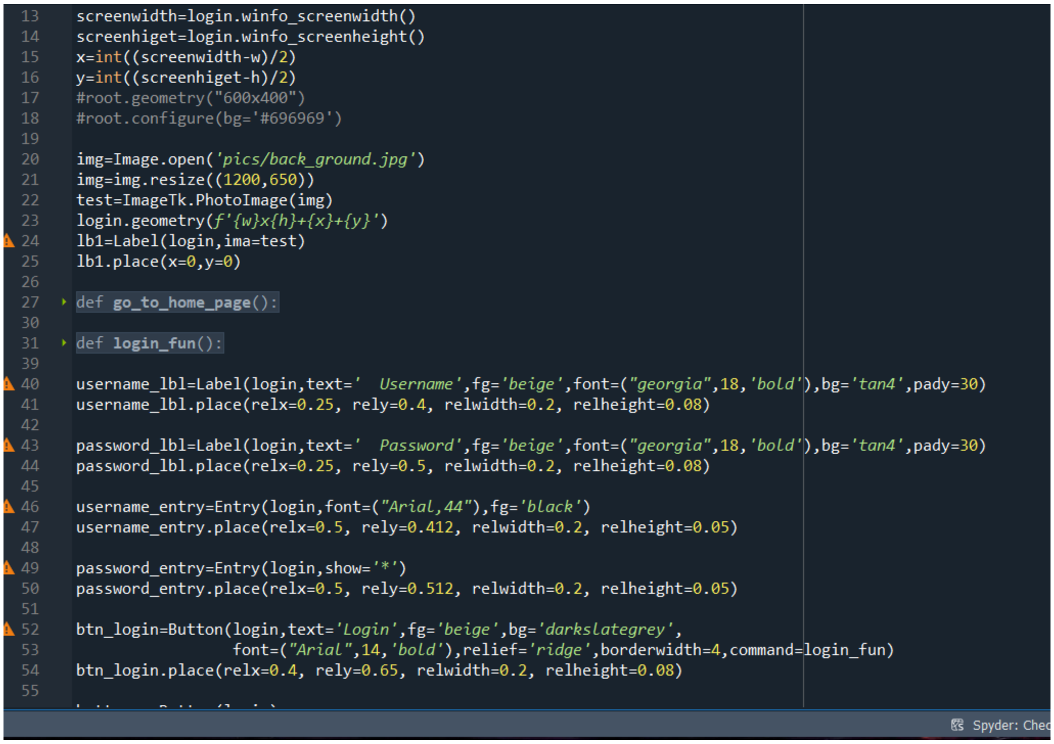


BookId	Book title	Author	Genre

~~BookId~~	~~Book title~~	Phone	Name	BorrowedDate	Due Date

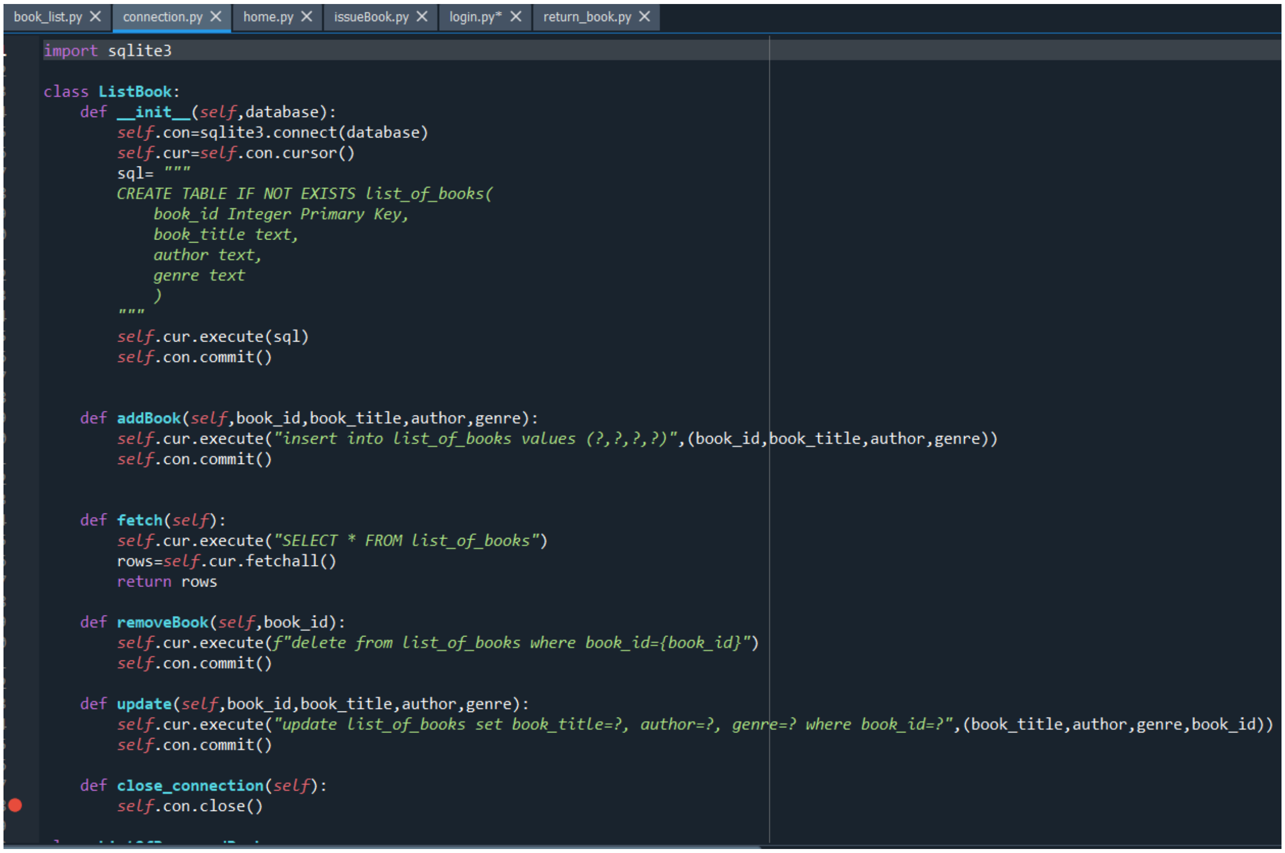
BookId	Phone

Code:

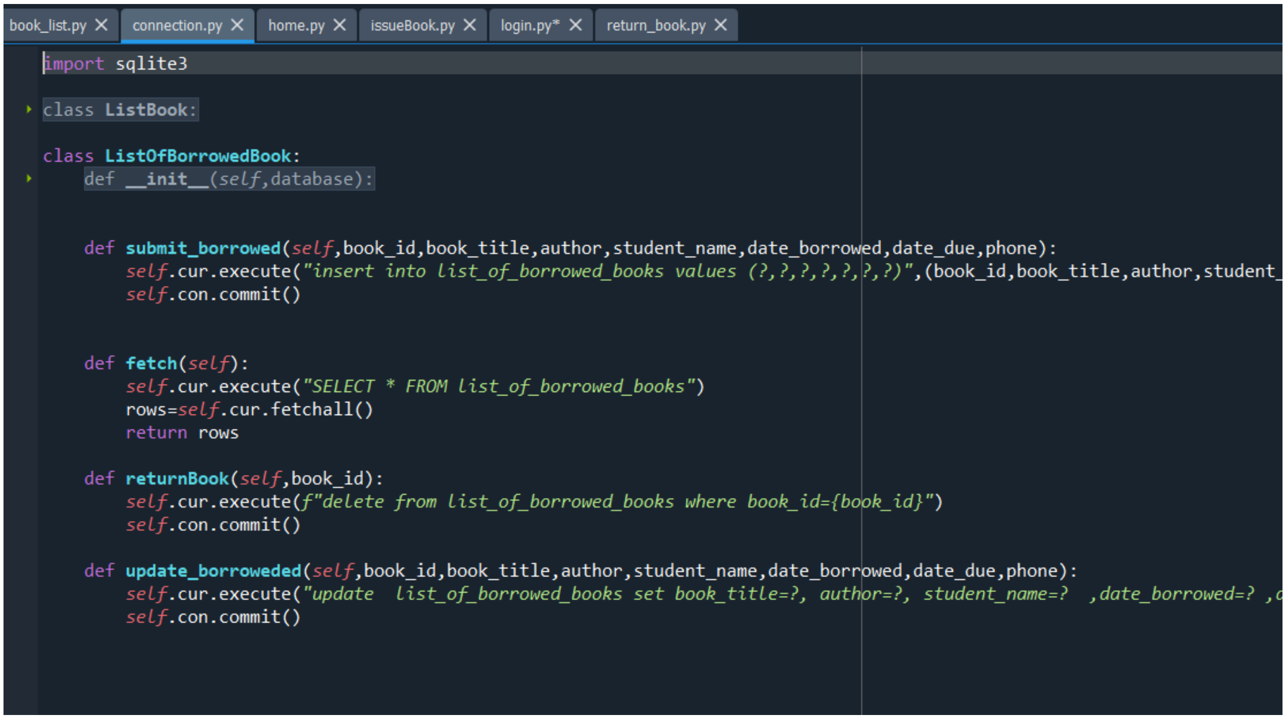


```
screenwidth=login.winfo_screenwidth()
screenhiget=login.winfo_screenheight()
x=int((screenwidth-w)/2)
y=int((screenhiget-h)/2)
#root.geometry("600x400")
#root.configure(bg='#696969')

img=Image.open('pics/back_ground.jpg')
img=img.resize((1200,650))
test=ImageTk.PhotoImage(img)
login.geometry(f'{w}x{h}+{x}+{y}')
lb1=Label(login,ima=test)
lb1.place(x=0,y=0)

def go_to_home_page():

def login_fun():

username_lbl=Label(login,text='  Username',fg='beige',font=("georgia",18,'bold'),bg='tan4',pady=30)
username_lbl.place(relx=0.25, rely=0.4, relwidth=0.2, relheight=0.08)

password_lbl=Label(login,text='  Password',fg='beige',font=("georgia",18,'bold'),bg='tan4',pady=30)
password_lbl.place(relx=0.25, rely=0.5, relwidth=0.2, relheight=0.08)

username_entry=Entry(login,font=("Arial,44"),fg='black')
username_entry.place(relx=0.5, rely=0.412, relwidth=0.2, relheight=0.05)

password_entry=Entry(login,show='*')
password_entry.place(relx=0.5, rely=0.512, relwidth=0.2, relheight=0.05)

btn_login=Button(login,text='Login',fg='beige',bg='darkslategrey',
                 font=("Arial",14,'bold'),relief='ridge',borderwidth=4,command=login_fun)
btn_login.place(relx=0.4, rely=0.65, relwidth=0.2, relheight=0.08)

```

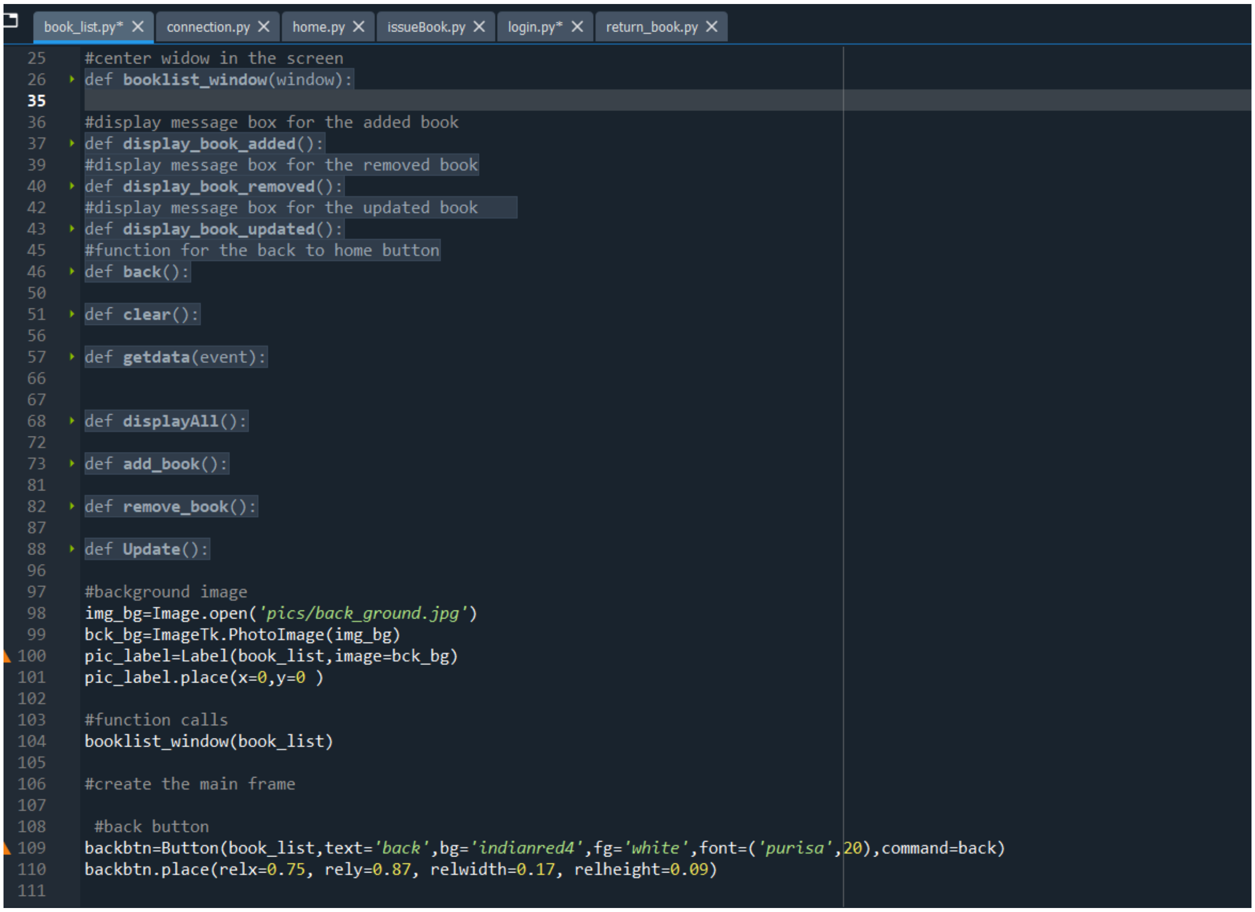
Spyder: Chec



```
import sqlite3

class ListBook:
    def __init__(self,database):
        self.con=sqlite3.connect(database)
        self.cur=self.con.cursor()
        sql= """
        CREATE TABLE IF NOT EXISTS list_of_books(
            book_id Integer Primary Key,
            book_title text,
            author text,
            genre text
            )
        """
        self.cur.execute(sql)
        self.con.commit()


    def addBook(self,book_id,book_title,author,genre):
        self.cur.execute("insert into list_of_books values (?,?,?,?)",(book_id,book_title,author,genre))
        self.con.commit()


    def fetch(self):
        self.cur.execute("SELECT * FROM list_of_books")
        rows=self.cur.fetchall()
        return rows

    def removeBook(self,book_id):
        self.cur.execute(f"delete from list_of_books where book_id={book_id}")
        self.con.commit()

    def update(self,book_id,book_title,author,genre):
        self.cur.execute("update list_of_books set book_title=?, author=?, genre=? where book_id=?",(book_title,author,genre,book_id))
        self.con.commit()

    def close_connection(self):
        self.con.close()
```



```
import sqlite3

class ListBook:

class ListOfBorrowedBook:
    def __init__(self,database):


    def submit_borrowed(self,book_id,book_title,author,student_name,date_borrowed,date_due,phone):
        self.cur.execute("insert into list_of_borrowed_books values (?,?,?,?,?,?,?)",(book_id,book_title,author,student_
        self.con.commit()


    def fetch(self):
        self.cur.execute("SELECT * FROM list_of_borrowed_books")
        rows=self.cur.fetchall()
        return rows

    def returnBook(self,book_id):
        self.cur.execute(f"delete from list_of_borrowed_books where book_id={book_id}")
        self.con.commit()

    def update_borroweded(self,book_id,book_title,author,student_name,date_borrowed,date_due,phone):
        self.cur.execute("update  list_of_borrowed_books set book_title=?, author=?, student_name=?  ,date_borrowed=? ,d
        self.con.commit()
```



```
#center widow in the screen
def booklist_window(window):

#display message box for the added book
def display_book_added():
#display message box for the removed book
def display_book_removed():
#display message box for the updated book
def display_book_updated():
#function for the back to home button
def back():

def clear():

def getdata(event):


def displayAll():

def add_book():

def remove_book():

def Update():

#background image
img_bg=Image.open('pics/back_ground.jpg')
bck_bg=ImageTk.PhotoImage(img_bg)
pic_label=Label(book_list,image=bck_bg)
pic_label.place(x=0,y=0 )

#function calls
booklist_window(book_list)

#create the main frame

 #back button
backbtn=Button(book_list,text='back',bg='indianred4',fg='white',font=('purisa',20),command=back)
backbtn.place(relx=0.75, rely=0.87, relwidth=0.17, relheight=0.09)

```