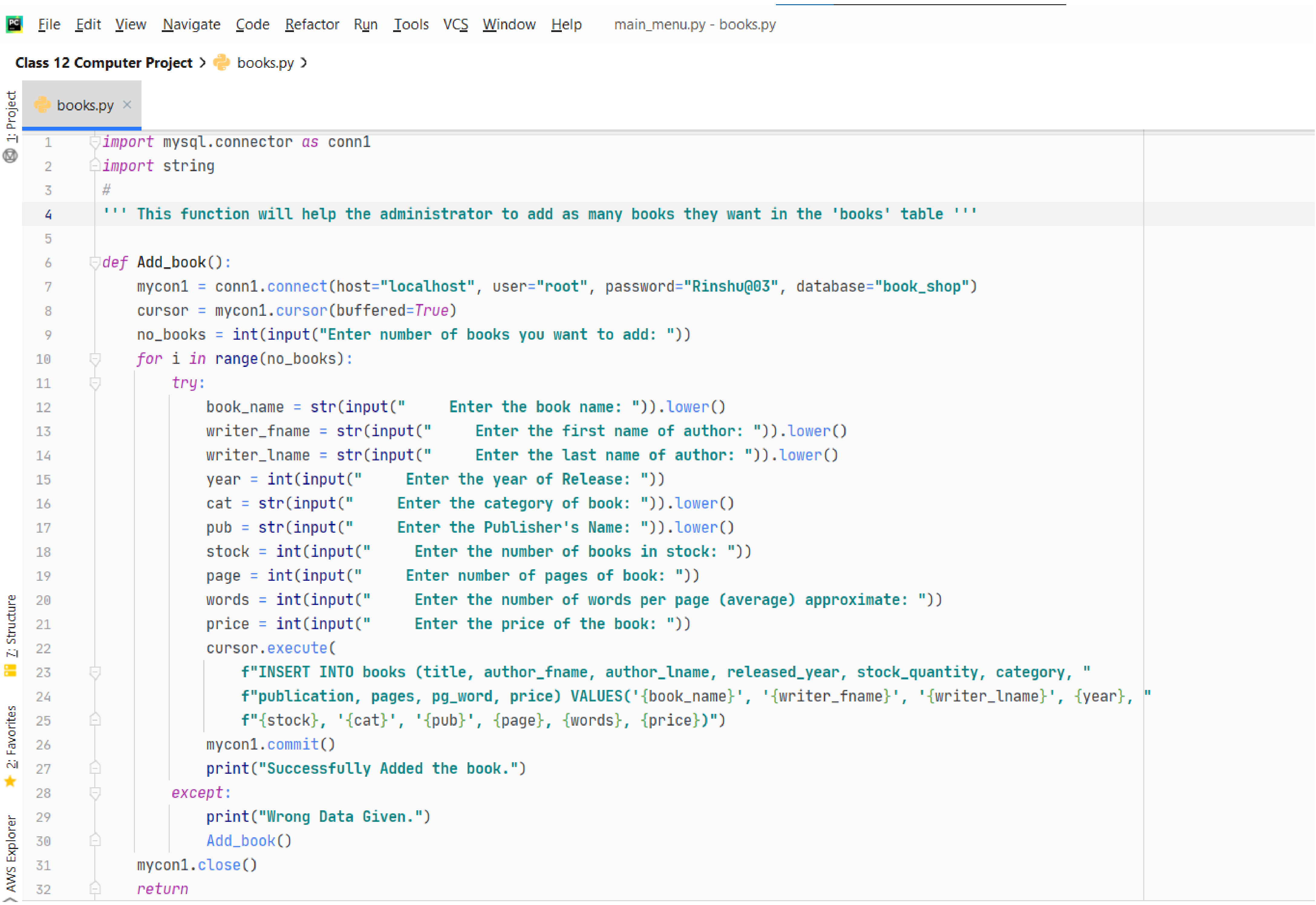
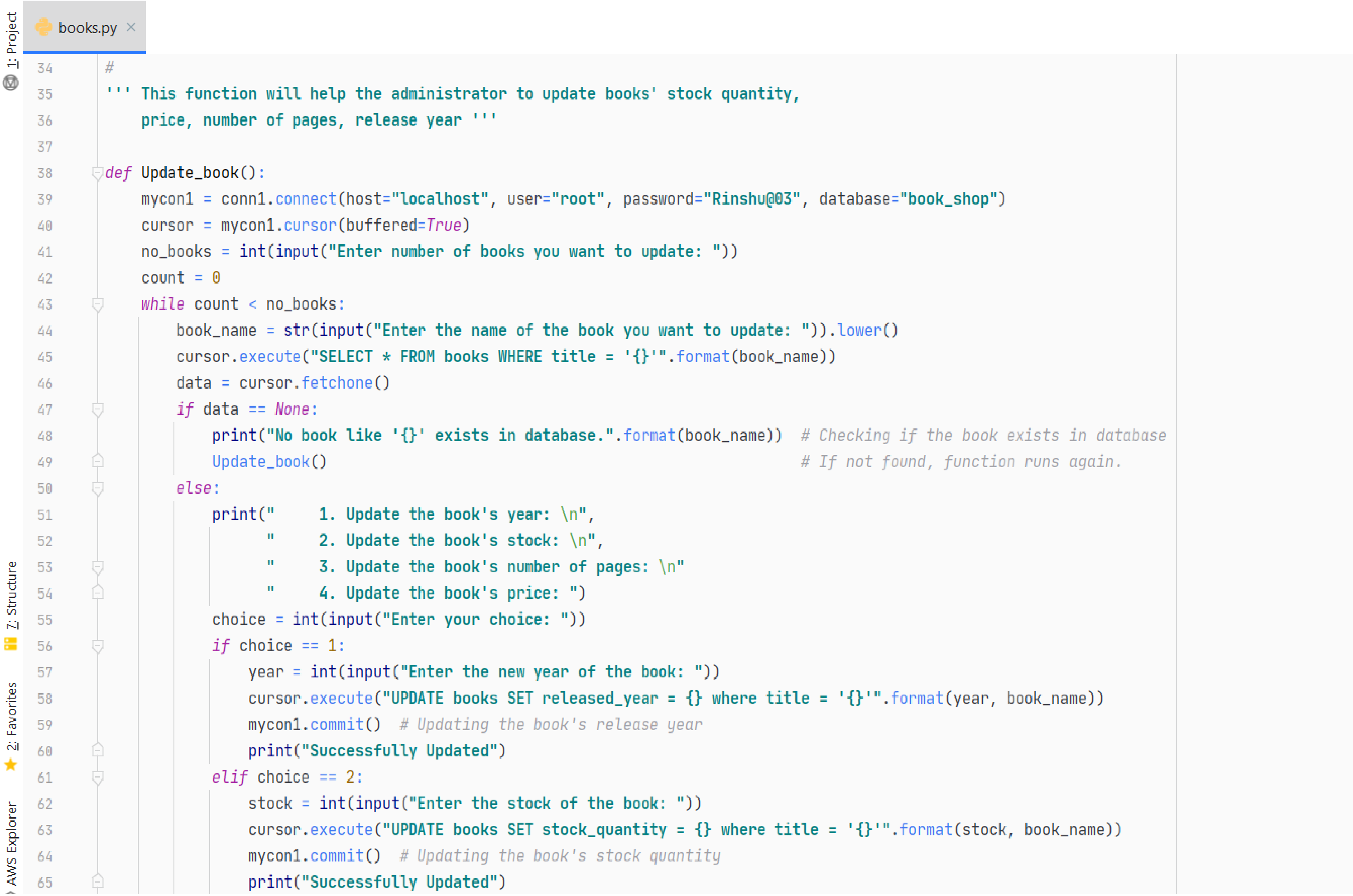
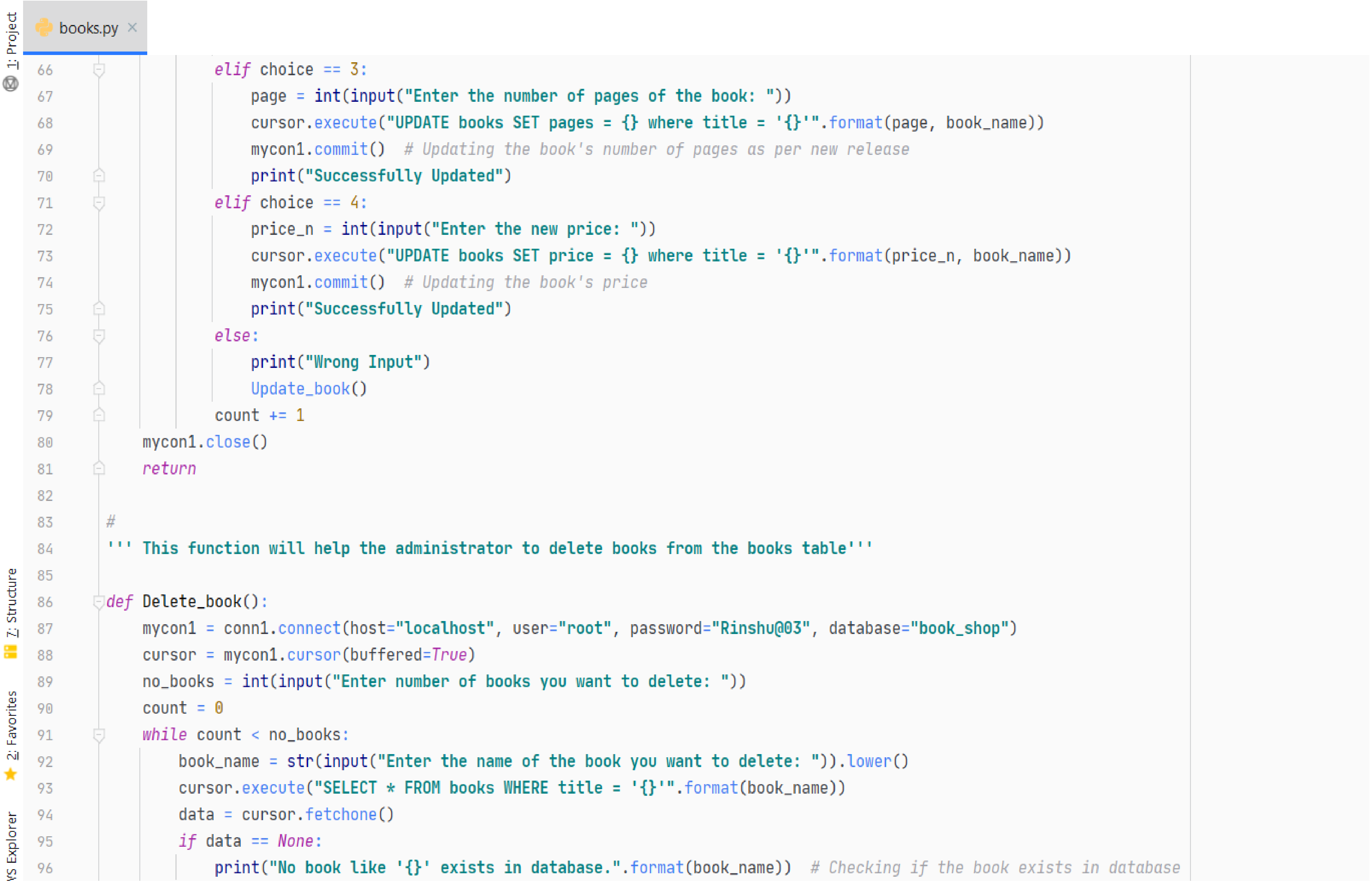


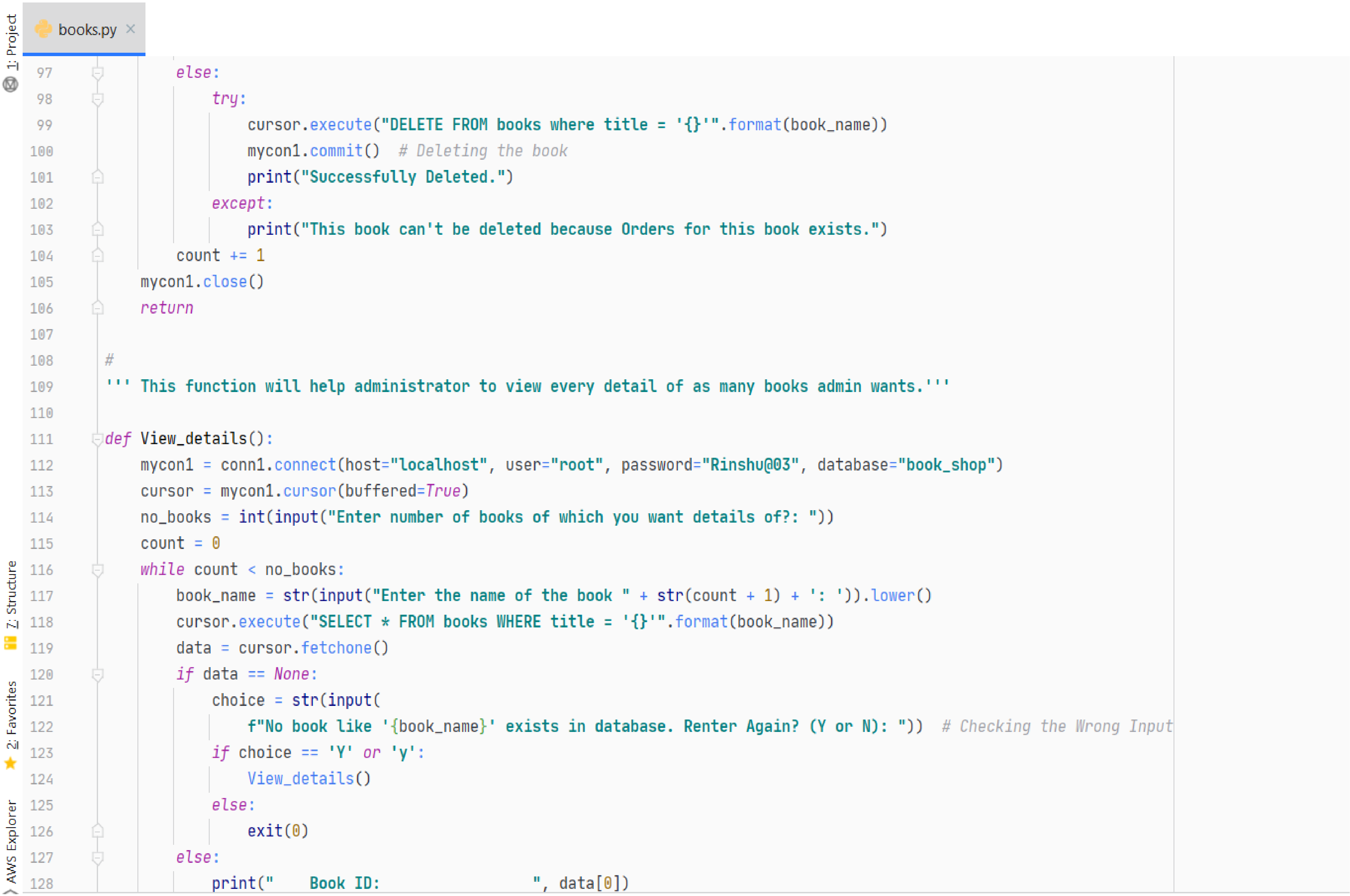
```python
import mysql.connector as conn1
import string
#
''' This function will help the administrator to add as many books they want in the 'books' table '''

def Add_book():
    mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon1.cursor(buffered=True)
    no_books = int(input("Enter number of books you want to add: "))
    for i in range(no_books):
        try:
            book_name = str(input("     Enter the book name: ")).lower()
            writer_fname = str(input("     Enter the first name of author: ")).lower()
            writer_lname = str(input("     Enter the last name of author: ")).lower()
            year = int(input("     Enter the year of Release: "))
            cat = str(input("     Enter the category of book: ")).lower()
            pub = str(input("     Enter the Publisher's Name: ")).lower()
            stock = int(input("     Enter the number of books in stock: "))
            page = int(input("     Enter number of pages of book: "))
            words = int(input("     Enter the number of words per page (average) approximate: "))
            price = int(input("     Enter the price of the book: "))
            cursor.execute(
                f"INSERT INTO books (title, author_fname, author_lname, released_year, stock_quantity, category, "
                f"publication, pages, pg_word, price) VALUES('{book_name}', '{writer_fname}', '{writer_lname}', {year}, "
                f"{stock}, '{cat}', '{pub}', {page}, {words}, {price})")
            mycon1.commit()
            print("Successfully Added the book.")
        except:
            print("Wrong Data Given.")
            Add_book()
    mycon1.close()
    return
```



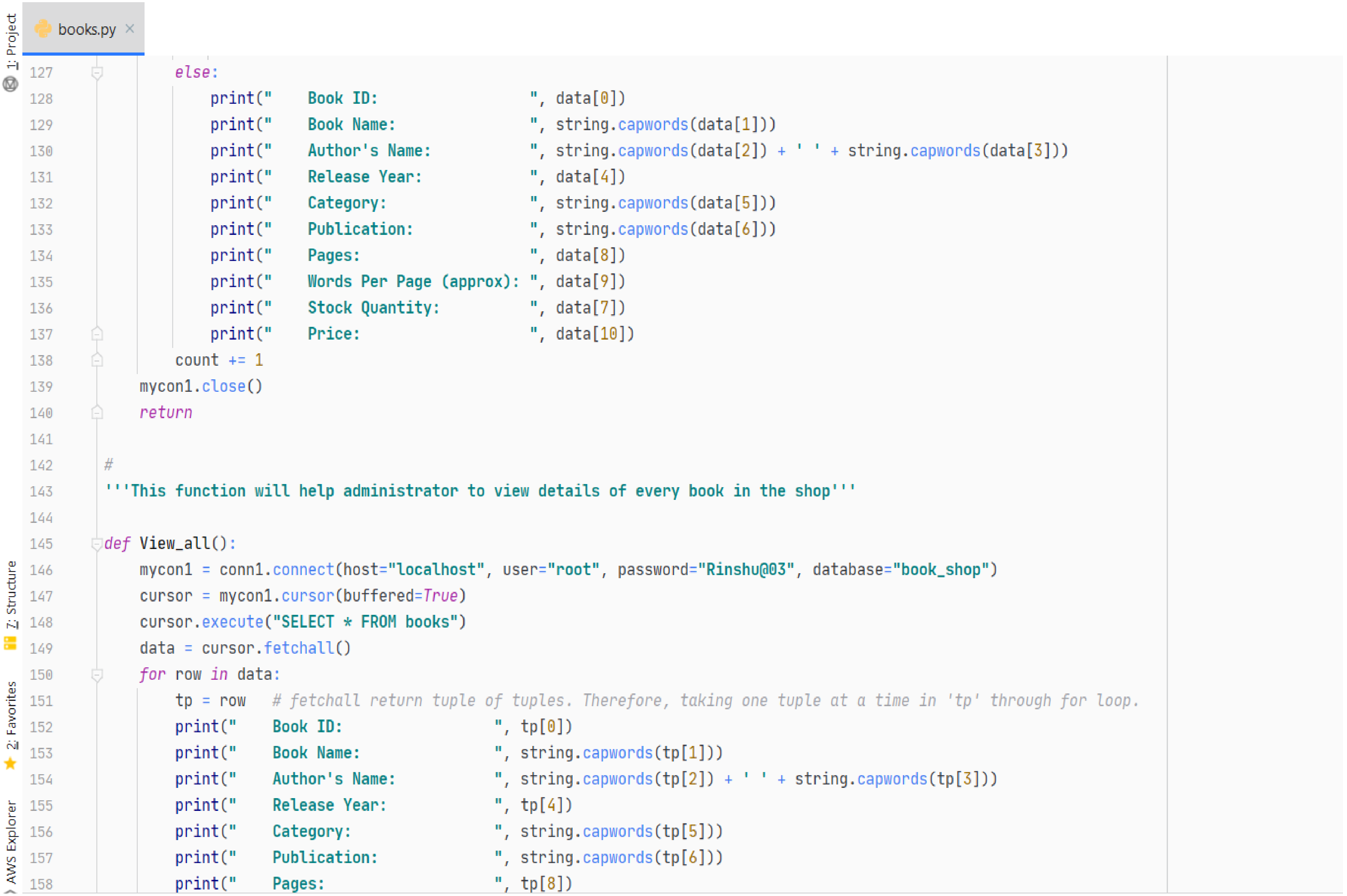
```python
#
''' This function will help the administrator to update books' stock quantity,
    price, number of pages, release year '''

def Update_book():
    mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon1.cursor(buffered=True)
    no_books = int(input("Enter number of books you want to update: "))
    count = 0
    while count < no_books:
        book_name = str(input("Enter the name of the book you want to update: ")).lower()
        cursor.execute("SELECT * FROM books WHERE title = '{}'".format(book_name))
        data = cursor.fetchone()
        if data == None:
            print("No book like '{}' exists in database.".format(book_name))  # Checking if the book exists in database
            Update_book()                                                     # If not found, function runs again.
        else:
            print("     1. Update the book's year: \n",
                  "     2. Update the book's stock: \n",
                  "     3. Update the book's number of pages: \n"
                  "     4. Update the book's price: ")
            choice = int(input("Enter your choice: "))
            if choice == 1:
                year = int(input("Enter the new year of the book: "))
                cursor.execute("UPDATE books SET released_year = {} where title = '{}'".format(year, book_name))
                mycon1.commit()  # Updating the book's release year
                print("Successfully Updated")
            elif choice == 2:
                stock = int(input("Enter the stock of the book: "))
                cursor.execute("UPDATE books SET stock_quantity = {} where title = '{}'".format(stock, book_name))
                mycon1.commit()  # Updating the book's stock quantity
                print("Successfully Updated")
```



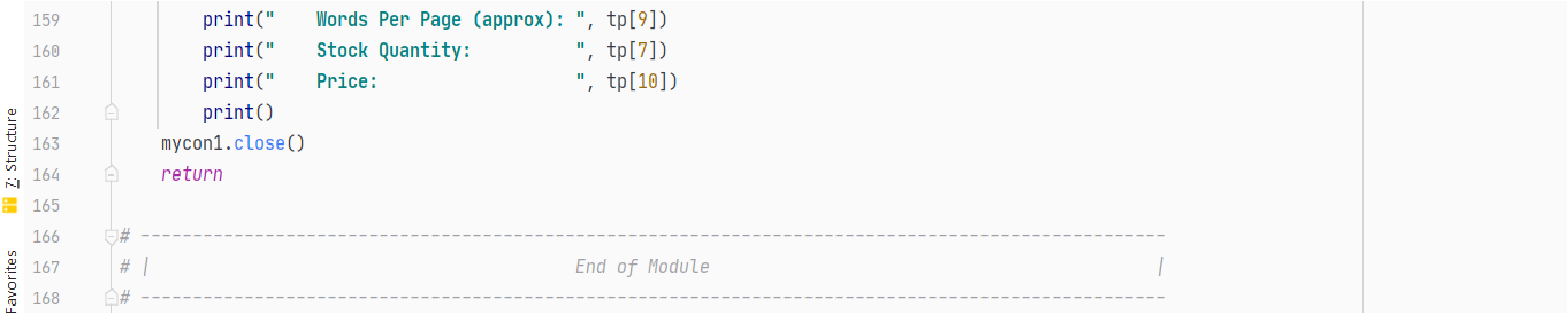
```python
            elif choice == 3:
                page = int(input("Enter the number of pages of the book: "))
                cursor.execute("UPDATE books SET pages = {} where title = '{}'".format(page, book_name))
                mycon1.commit()  # Updating the book's number of pages as per new release
                print("Successfully Updated")
            elif choice == 4:
                price_n = int(input("Enter the new price: "))
                cursor.execute("UPDATE books SET price = {} where title = '{}'".format(price_n, book_name))
                mycon1.commit()  # Updating the book's price
                print("Successfully Updated")
            else:
                print("Wrong Input")
                Update_book()
            count += 1
    mycon1.close()
    return

#
''' This function will help the administrator to delete books from the books table'''

def Delete_book():
    mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon1.cursor(buffered=True)
    no_books = int(input("Enter number of books you want to delete: "))
    count = 0
    while count < no_books:
        book_name = str(input("Enter the name of the book you want to delete: ")).lower()
        cursor.execute("SELECT * FROM books WHERE title = '{}'".format(book_name))
        data = cursor.fetchone()
        if data == None:
            print("No book like '{}' exists in database.".format(book_name))  # Checking if the book exists in database
```



```python
        else:
            try:
                cursor.execute("DELETE FROM books where title = '{}'".format(book_name))
                mycon1.commit()  # Deleting the book
                print("Successfully Deleted.")
            except:
                print("This book can't be deleted because Orders for this book exists.")
        count += 1
    mycon1.close()
    return

#
''' This function will help administrator to view every detail of as many books admin wants.'''

def View_details():
    mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon1.cursor(buffered=True)
    no_books = int(input("Enter number of books of which you want details of?: "))
    count = 0
    while count < no_books:
        book_name = str(input("Enter the name of the book " + str(count + 1) + ': ')).lower()
        cursor.execute("SELECT * FROM books WHERE title = '{}'".format(book_name))
        data = cursor.fetchone()
        if data == None:
            choice = str(input(
                f"No book like '{book_name}' exists in database. Renter Again? (Y or N): "))  # Checking the Wrong Input
            if choice == 'Y' or 'y':
                View_details()
            else:
                exit(0)
        else:
            print("    Book ID:                 ", data[0])
```



```
        else:
            print("    Book ID:                 ", data[0])
            print("    Book Name:               ", string.capwords(data[1]))
            print("    Author's Name:           ", string.capwords(data[2]) + ' ' + string.capwords(data[3]))
            print("    Release Year:            ", data[4])
            print("    Category:                ", string.capwords(data[5]))
            print("    Publication:             ", string.capwords(data[6]))
            print("    Pages:                   ", data[8])
            print("    Words Per Page (approx): ", data[9])
            print("    Stock Quantity:          ", data[7])
            print("    Price:                   ", data[10])
        count += 1
    mycon1.close()
    return

#
'''This function will help administrator to view details of every book in the shop'''

def View_all():
    mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon1.cursor(buffered=True)
    cursor.execute("SELECT * FROM books")
    data = cursor.fetchall()
    for row in data:
        tp = row   # fetchall return tuple of tuples. Therefore, taking one tuple at a time in 'tp' through for loop.
        print("    Book ID:                 ", tp[0])
        print("    Book Name:               ", string.capwords(tp[1]))
        print("    Author's Name:           ", string.capwords(tp[2]) + ' ' + string.capwords(tp[3]))
        print("    Release Year:            ", tp[4])
        print("    Category:                ", string.capwords(tp[5]))
        print("    Publication:             ", string.capwords(tp[6]))
        print("    Pages:                   ", tp[8])
```



```
        print("    Words Per Page (approx): ", tp[9])
        print("    Stock Quantity:          ", tp[7])
        print("    Price:                   ", tp[10])
        print()
    mycon1.close()
    return

# ----------------------------------------------------------------------------------------------------
# |                                         End of Module                                           |
# ----------------------------------------------------------------------------------------------------
```