



Class 12 Computer Project > books.py >

Project
1: Project

books.py ×

```
1 import mysql.connector as conn1
2 import string
3 #
4 ''' This function will help the administrator to add as many books they want in the 'books' table '''
5
6 def Add_book():
7     mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
8     cursor = mycon1.cursor(buffered=True)
9     no_books = int(input("Enter number of books you want to add: "))
10    for i in range(no_books):
11        try:
12            book_name = str(input("        Enter the book name: ")).lower()
13            writer_fname = str(input("        Enter the first name of author: ")).lower()
14            writer_lname = str(input("        Enter the last name of author: ")).lower()
15            year = int(input("        Enter the year of Release: "))
16            cat = str(input("        Enter the category of book: ")).lower()
17            pub = str(input("        Enter the Publisher's Name: ")).lower()
18            stock = int(input("        Enter the number of books in stock: "))
19            page = int(input("        Enter number of pages of book: "))
20            words = int(input("        Enter the number of words per page (average) approximate: "))
21            price = int(input("        Enter the price of the book: "))
22            cursor.execute(
23                f"INSERT INTO books (title, author_fname, author_lname, released_year, stock_quantity, category, "
24                f"publication, pages, pg_word, price) VALUES('{book_name}', '{writer_fname}', '{writer_lname}', {year}, "
25                f"{stock}, '{cat}', '{pub}', {page}, {words}, {price})")
26            mycon1.commit()
27            print("Successfully Added the book.")
28        except:
29            print("Wrong Data Given.")
30            Add_book()
31    mycon1.close()
32    return
```

Z: Structure

2: Favorites

AWS Explorer

```
34 #
35 ''' This function will help the administrator to update books' stock quantity,
36     price, number of pages, release year '''
37
38 def Update_book():
39     mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
40     cursor = mycon1.cursor(buffered=True)
41     no_book = int(input("Enter number of books you want to update: "))
42     count = 0
43     while count < no_book:
44         book_name = str(input("Enter the name of the book you want to update: ")).lower()
45         cursor.execute("SELECT * FROM books WHERE title = '{}'.format(book_name)")
46         data = cursor.fetchone()
47         if data == None:
48             print("No book like '{}' exists in database.".format(book_name)) # Checking if the book exists in database
49             Update_book() # If not found, function runs again.
50         else:
51             print("    1. Update the book's year: \n",
52                   "    2. Update the book's stock: \n",
53                   "    3. Update the book's number of pages: \n"
54                   "    4. Update the book's price: ")
55             choice = int(input("Enter your choice: "))
56             if choice == 1:
57                 year = int(input("Enter the new year of the book: "))
58                 cursor.execute("UPDATE books SET released_year = {} where title = '{}'.format(year, book_name)")
59                 mycon1.commit() # Updating the book's release year
60                 print("Successfully Updated")
61             elif choice == 2:
62                 stock = int(input("Enter the stock of the book: "))
63                 cursor.execute("UPDATE books SET stock_quantity = {} where title = '{}'.format(stock, book_name)")
64                 mycon1.commit() # Updating the book's stock quantity
65                 print("Successfully Updated")
```

```
66 elif choice == 3:
67     page = int(input("Enter the number of pages of the book: "))
68     cursor.execute("UPDATE books SET pages = {} where title = '{}'.format(page, book_name))
69     mycon1.commit() # Updating the book's number of pages as per new release
70     print("Successfully Updated")
71 elif choice == 4:
72     price_n = int(input("Enter the new price: "))
73     cursor.execute("UPDATE books SET price = {} where title = '{}'.format(price_n, book_name))
74     mycon1.commit() # Updating the book's price
75     print("Successfully Updated")
76 else:
77     print("Wrong Input")
78     Update_book()
79     count += 1
80 mycon1.close()
81 return
82
83 #
84 ''' This function will help the administrator to delete books from the books table'''
85
86 def Delete_book():
87     mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
88     cursor = mycon1.cursor(buffered=True)
89     no_books = int(input("Enter number of books you want to delete: "))
90     count = 0
91     while count < no_books:
92         book_name = str(input("Enter the name of the book you want to delete: ")).lower()
93         cursor.execute("SELECT * FROM books WHERE title = '{}'.format(book_name))
94         data = cursor.fetchone()
95         if data == None:
96             print("No book like '{}' exists in database.".format(book_name)) # Checking if the book exists in database
```

```
97     else:
98         try:
99             cursor.execute("DELETE FROM books where title = '{}'.format(book_name))
100             mycon1.commit() # Deleting the book
101             print("Successfully Deleted.")
102         except:
103             print("This book can't be deleted because Orders for this book exists.")
104     count += 1
105     mycon1.close()
106     return
107
108     #
109     ''' This function will help administrator to view every detail of as many books admin wants.'''
110
111     def View_details():
112         mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
113         cursor = mycon1.cursor(buffered=True)
114         no_books = int(input("Enter number of books of which you want details of?: "))
115         count = 0
116         while count < no_books:
117             book_name = str(input("Enter the name of the book " + str(count + 1) + ': ')).lower()
118             cursor.execute("SELECT * FROM books WHERE title = '{}'.format(book_name))
119             data = cursor.fetchone()
120             if data == None:
121                 choice = str(input(
122                     f"No book like '{book_name}' exists in database. Renter Again? (Y or N): ")) # Checking the Wrong Input
123                 if choice == 'Y' or 'y':
124                     View_details()
125                 else:
126                     exit(0)
127             else:
128                 print("    Book ID:          ", data[0])
```

```
127     else:
128         print("    Book ID:           ", data[0])
129         print("    Book Name:          ", string.capwords(data[1]))
130         print("    Author's Name:      ", string.capwords(data[2]) + ' ' + string.capwords(data[3]))
131         print("    Release Year:       ", data[4])
132         print("    Category:           ", string.capwords(data[5]))
133         print("    Publication:        ", string.capwords(data[6]))
134         print("    Pages:              ", data[8])
135         print("    Words Per Page (approx): ", data[9])
136         print("    Stock Quantity:     ", data[7])
137         print("    Price:              ", data[10])
138     count += 1
139 mycon1.close()
140 return
141
142 #
143 '''This function will help administrator to view details of every book in the shop'''
144
145 def View_all():
146     mycon1 = conn1.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
147     cursor = mycon1.cursor(buffered=True)
148     cursor.execute("SELECT * FROM books")
149     data = cursor.fetchall()
150     for row in data:
151         tp = row # fetchall return tuple of tuples. Therefore, taking one tuple at a time in 'tp' through for loop.
152         print("    Book ID:           ", tp[0])
153         print("    Book Name:          ", string.capwords(tp[1]))
154         print("    Author's Name:      ", string.capwords(tp[2]) + ' ' + string.capwords(tp[3]))
155         print("    Release Year:       ", tp[4])
156         print("    Category:           ", string.capwords(tp[5]))
157         print("    Publication:        ", string.capwords(tp[6]))
158         print("    Pages:              ", tp[8])
```

```
159     print("    Words Per Page (approx): ", tp[9])
160     print("    Stock Quantity:          ", tp[7])
161     print("    Price:                    ", tp[10])
162     print()
163     mycon1.close()
164     return
165
166 # -----
167 # |                                     End of Module                                     |
168 # -----
```