```
🕽 <u>1</u>: Project
```

orders.pv X

```
# Inbuilt Modules
      import smtplib
 2
      from string import capwords
 3
      from time import sleep
 4
      import datetime
 5
      # Project Modules
 7
      import mysql.connector as mycam
 8
      import config
 9
10
11
        ______
12
                             Below Functions are Common
13
      # ------ #
14
15
     ^# -----
16
     ♥''' This function will allow every user in database to place order for any book if it exists in database.
17
         User is prompted every time to place another order if he/she wants.
18
         When user says 'n' or 'no' function are called from other modules to generate invoice and invoice number.'''
19
20
      def Place_ord(username):
21
         ord_id_ls = []
22
         cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
         cursor = cam.cursor(buffered=True)
24
         cursor.execute("SET SQL_SAFE_UPDATES = 0")
25
26
         def work():
27
             print()
28
            book_name = str(input('Enter the name of the book ' + ': ')).lower()
            cursor.execute("SELECT * FROM books WHERE title='{}'".format(book_name))
30
            data = cursor.fetchone()
31
             if data == None:
```

```
Project
     orders.py X
   31
                   data = cursor.tetchone()
÷
   32
                   if data == None:
                       print('Sorry, No book named', book_name, 'found.')
   33
   34
                   else:
                                  Book named , {capwords(book_name)}, found details are as follows:-")
   35
                       print(f"
                                                            ", capwords(data[2]) + ' ' + capwords(data[3]))
   36
                       print("
                                  Author's name:
                                  Released Year:
                                                            ", data[4])
   37
                       print("
                                  Number of Pages:
                                                            ", data[8])
                       print("
   38
                                  Words Per Page (Approx): ", data[9])
                       print("
   39
                                                            ", capwords(data[6]))
   40
                       print("
                                   Publication:
                                  Category of Book:
                                                            ", capwords(data[5]))
   41
                       print("
                                  Available quantity:
                                                            ", data[7])
   42
                       print("
                                  Price of the book:
                                                            ", data[10])
                       print("
   43
                       choice = str(input("Do you want to order this Book ? "))
   44
   45
                       if choice in ['yes', 'Yes', 'YES', 'y', 'Y']:
   46
                           quantity = int(input('Enter the QUANTITY you want to order: '))
   47
                           print('
                                                 Checking Stock, please wait...')
   48
                           sleep(2)
   49
   50
                           if quantity <= data[7]: # Checking if pieces specified to be bought are present or not for the specified book.
   51
                                                     Enough quantity found')
   52
                               print('
                               print('Please fill up your details below: ')
   53
   54
                                                       Enter Name: '))
                               name = str(input('
                               address = str(input('
                                                          Enter Address: '))
                               dist = str(input('
                                                       Enter District: '))
                               city = str(input('
                                                       Enter City: '))
                               stu = str(input('
                                                      Enter State/UT: '))
Explorer
                               pin = int(input("
                                                      Enter Pincode: "))
                               country = str(input('
                                                          Enter Country: '))
  61
AWS
                               phone = int(input('
                                                        Enter Phone number: '))
   62
```

```
Project
     orders.py X
<del>∵</del>I 63
                                # Email Address is fetched from the accounts table. This email address will be used to send email
                                cursor.execute(f"SELECT email FROM accounts where username = '{username}'")
    65
                                email = cursor.fetchone()
    66
                                amount = data[10] * quantity
   67
                                cursor.execute(f"SELECT ID FROM accounts WHERE username = '{username}'")
    68
                                custom id = cursor.fetchone()
    69
                                print(' Now we came at the final step of the process...')
   70
                                                    Total Amount to be paid is: ", amount)
                                print("
   71
                                                    No Delivery Charges Applicable, i.e. Delivery Free")
   72
                                print("
                                confirm = str(input(' Are you sure you want to confirm this order? '))
   73
   74
                                if confirm in ['yes', 'Yes', 'YES', 'y', 'Y']:
   75
                                    details = [name, address, dist, city, stu, pin, country, phone, custom_id]
   76
   77
                                    cursor.execute(
                                        f"UPDATE books SET stock_quantity = stock_quantity - {quantity} WHERE title = '{book_name}'")
   78
                                    cam.commit()
   79
                                    cursor.execute(
    80
                                        f"INSERT INTO orders "
   81
                                        f"(customer_name, address, district, city, state_ut, pincode, country, e_mail, phone_no, pieces, bookname, "
   82
                                        f"user_name, order_type) "
   83
                                        f"VALUES ('{name}', '{address}', '{dist}', '{city}', '{stu}', {pin}, '{country}', '{email[0]}', {phone}, {quantity}, "
   84
                                        f"'{book_name}', '{username}', 'new')")
   85
                                    cam.commit()
    86
                                    cursor.execute(
   88
                                        f"SELECT ord_no, order_dt FROM orders WHERE customer_name = '{name}' and phone_no = {phone} DRDER BY ord_no DESC")
ii 89
                                    data2 = cursor.fetchone()
                                    ord_id_ls.append(data2[0])
   90
                                    print("Processing your order....")
   91
   92
                                    sleep(2)
                                    print()
   93
                                    print("
                                                          Order Placed Successfully.")
   94
```

```
1: Project
      orders.py X
                                     print()
                                                            Order Placed Successfully.")
   94
                                     print("
                                 else:
   95
                                     print('Your order has been discarded')
   96
   97
                             else:
   98
                                 print('Sorry', data[7], 'pieces are left')
   99
                                 print('Thanks for visiting, keep shopping :)')
  100
                        elif choice in ['no', 'No', 'NO', 'n', 'N']:
  101
                             print('Okay.')
  102
                        else:
  103
                             print('Invalid Input')
  104
  105
                    print()
  106
                    ch = str(input("Want to place more orders? (y/n): ")).lower()
  107
                    if ch in ['y', 'yes']:
  108
                        work()
  109
                    else:
  110
                        details.append(email[0])
  111
                        details.append(username)
  112
Structure
                        inv = config.Inv_no_gen(username)
  113
  114
                        config.Pdf_generate(details, ord_id_ls, inv[0], inv[1])
                        print("Thanks for visiting us :)")
  115
                        print()
  116
2: Favorites
  117
                work()
  118
                return
  119
  120
           '''This function will be used only if the order is specified for return/replace or order is cancelled
  121
AWS Explorer
  122
               For that, message and subject or email is defined in their respective functions.
  123
               Sender's Email Address and Password is given in config module of the project
               An email will be sent to the email address of the user with the details.'''
  124
```

```
Project
      orders.py X
<del>~</del> 125
           Idef Send_email(email, subject, msg):
   126
                try:
   127
                    server = smtplib.SMTP('smtp.gmail.com:587')
   128
   129
                    server.ehlo()
                    server.starttls()
   130
                    server.login(config.EMAIL_ADDRESS, config.PASSWORD)
   131
                    final_message = 'Subject: {}\n\n{}'.format(subject, msg)
   132
   133
                    server.sendmail(config.EMAIL_ADDRESS, email, final_message)
                    server.quit()
   134
                    print(f"An email with all details is sent to you on your E-mail Address {email}")
   135
                except:
   136
                    print("Email failed to send.")
   137
                return
   138
   139
   140
           !''This function will allow the user to return or replace the order within 7 days after the delivery date.
   141
               Else a message will be displayed that the order is not eligible for passing the Return/Replace Policy.
   142
               By default, the delivery date is the date 7 days after the date of placing the order'''
   143
   144
   145
           def Return_replace(username):
   146
                cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
                cursor = cam.cursor(buffered=True)
  147
                cursor.execute(f"SELECT ord_no FROM orders WHERE user_name = '{username}'")
   148
Favorites
                record = cursor.fetchall()
   149
   150
                L = []
<sup>™</sup> 151
                if not record:
                    print('No order to return or replace')
   152
                else:
   153
AWS Explorer
   154
                    cursor.execute(
```

155

156

f"ORDER by ord\_no DESC")

f"SELECT \* from orders where user\_name = '{username}' and DATEDIFF(NOW(), order\_dt) >= 7 and DATEDIFF(NOW(), order\_dt) <= 14 "

```
Project
      orders.py X
                    data = cursor.fetchall()
   157
                    if not data:
   158
                        print('None of your order is eligible to return or replace.')
   159
                    else:
   160
                        print('We have found', len(data), 'orders have been delivered to you')
   161
                        print('List of the orders that have been delivered to you are as follows:-')
   162
                        for i in data:
   163
                            L.append(i[0])
   164
                            print('
                                                     : ', i[11])
                                      Book Name
   165
                            print('
                                                     : ', i[12])
                                      Order Date
   166
                            print(' Order ID
                                                      : ', i[0])
   167
                                      Customer Name : ', i[1])
                            print('
   168
                            print(f" Location
                                                     : {i[2]}, {i[3]}, {i[4]}, {i[5]}, {i[6]}, {i[7]}")
   169
                            print('
                                                      : ', i[8])
                                      Email
  170
                                                      : ', i[9])
                            print('
                                      Phone no
  171
                                                      : ', i[10])
  172
                            print('
                                      No of Books
                            print()
  173
  174
                        choice = str(input('Do you want to return or replace any of the above orders <math>(y/n) ')).lower()
   175
   176
                        def work(c):
   177
                            if c == 'v':
FI 178
  179
                                ID = int(input('Enter the order ID for which you want to return or replace: '))
                                if ID not in L:
   180
Favorites
                                    print('No order ID ', ID, ' matched from the orders that have been delivered to you')
  181
  182
                                else:
                                    print('You have chosen order ID: ', ID)
   183
                                    print('Few details of your order are as follows: ')
   184
                                    print()
AWS Explorer
  185
  186
                                    cursor.execute(f"SELECT * FROM orders WHERE ord_no = {ID}")
                                    data2 = cursor.fetchone()
   187
                                    print('
                                              Book name: '. data2[11])
   188
```

```
1: Project
      orders.py X
   165
                            brint(.
                                      воок маше
                                                      : ', 1[11])
: ', i[12])
                            print('
                                      Order Date
                                                      : ', i[0])
   167
                            print('
                                      Order ID
                            print('
                                      Customer Name : ', i[1])
   168
                            print(f"
                                                      : {i[2]}, {i[3]}, {i[4]}, {i[5]}, {i[6]}, {i[7]}")
                                       Location
   169
                            print('
                                                      : ', i[8])
                                       Email
   170
                            print('
                                                      : ', i[9])
   171
                                      Phone no
                            print('
                                                      : ', i[10])
   172
                                      No of Books
                            print()
   173
   174
                        choice = str(input('Do you want to return or replace any of the above orders <math>(y/n) ')).lower()
  175
   176
                        def work(c):
   177
                            if c == 'y':
   178
  179
                                ID = int(input('Enter the order ID for which you want to return or replace: '))
                                if ID not in L:
   180
                                    print('No order ID ', ID, ' matched from the orders that have been delivered to you')
   181
   182
                                else:
                                    print('You have chosen order ID: ', ID)
   183
                                    print('Few details of your order are as follows: ')
   184
   185
                                    print()
                                    cursor.execute(f"SELECT * FROM orders WHERE ord_no = {ID}")
   186
   187
                                    data2 = cursor.fetchone()
                                               Book name: ', data2[11])
   188
                                    print('
5: Favorites
190
191
                                               Quantity: ', data2[10])
                                    print('
                                    print('
                                               Order date: ', data2[12])
                                     print()
192
                                    print('By SNT.bookshop Return and Replace policies, you have below two options')
                                    print('1. Return')
   193
AWS Explorer
   194
                                    print('2. Replace')
   195
                                    opt = int(input('Enter the option number that you want to select: '))
   196
```

```
orders.py X
<del>~</del>I 197
   198
                                    if opt == 1:
  199
                                        print('You have entered under our Return policy')
                                        confirm = str(input('Are you sure you want to return this order? (y/n) ')).lower()
  200
                                        if confirm in ['v', 'ves']:
  201
                                            print(' Processing the Return')
  202
                                            msg = f"Your request for returning your order has been accepted. Some details for it are as follows: \n" \
  203
                                                  f"Book Name: {capwords(data2[11])} \n" \
  204
                                                  f"Order ID: {data2[0]} \n" \
  205
                                                  f"Name: {data2[1]} \n" \
  206
                                                  f"Address: {data2[2]} \n" \
  207
                                                  f"District: {data2[3]} \n" \
  208
                                                  f"City: {data2[4]} \n" \
  209
                                                  f"State: {data2[5]} \n" \
  210
                                                  f"Pincode: {data2[6]} \n" \
  211
  212
                                                  f"Country: {data2[7]} \n" \
                                                  f"Phone: {data2[9]} \n" \
  213
                                                  f"Pieces Ordered: {data2[10]} \n" \
  214
                                                  f"Order Date & Time: {data2[12]} \n"
   215
                                            sub = f"Information of Returning Order on S & T Bookshop"
  216
  217
                                            Send_email(data2[8], sub, msq) # Calling Function to send email
  218
                                            # Real Email is sent here.
                                            # Email is sent actually to the mailbox if the given email really exists
219
   220
                                            cursor.execute(
  221
                                                f"INSERT INTO orders "
  222
                                                f"(customer_name, address, district, city, state_ut, pincode, country, e_mail, phone_no, pieces, bookname, "
NI 223
                                                f"user_name, order_type) VALUES ('{data2[1]}', '{data2[2]}', '{data2[3]}', '{data2[4]}', '{data2[5]}', "
                                                f"{data2[6]}, '{data2[7]}', '{data2[8]}', {data2[9]}, {data2[10]}, "
  224
                                                f"'{data2[11]}', '{username}', 'return')")
  225
   226
                                            cam.commit()
  227
                                            cursor.execute(f"DELETE FROM orders WHERE ord_no = {ID}")
                                            cam.commit()
  228
```

```
Project
      orders.py X
229
                                             print('
                                                       Successfully requested for return to your order.')
                                             print(
   230
                                                     Our courier boy will come within the next day to collect the package after verifying its condition,
   231
                                                     be ready with the packed book')
   232
                                                       Money will be refund to your account within 2-3 days')
   233
                                             print('
   234
                                             return
   235
                                         else:
   236
                                             print('
                                                       Return process cancelled')
   237
   238
                                     elif opt == 2:
                                         print('You have entered under our Replace policy.')
   239
                                         confirm = str(input('Are you sure you want to replace for this order? (y/n) ')).lower()
   240
   241
                                         if confirm in ['y', 'yes']:
   242
                                             msg = f"Your request for replacing your order has been accepted. Some details for it are as follows: \n" \
   243
                                                   f"Book Name: {capwords(data2[11])} \n" \
   244
                                                   f"Order ID: {data2[0]} \n" \
   245
                                                   f"Name: {data2[1]} \n" \
   246
                                                   f"Address: {data2[2]} \n" \
   247
                                                   f"District: {data2[3]} \n" \
   248
                                                   f"City: {data2[4]} \n" \
   249
                                                   f"State: {data2[5]} \n" \
~i 250
   251
                                                   f"Pincode: {data2[6]} \n" \
                                                   f"Country: {data2[7]} \n" \
Favorites
                                                   f"Phone: {data2[9]} \n" \
   253
                                                   f"Pieces Ordered: {data2[10]} \n" \
   254
                                                   f"Order Date & Time: {data2[12]} \n" \
   255
                                                   f"Delivery will be provided within 7 working days. \n" \
   256
                                                   f"Sorry for the inconvenience :)"
  257
AWS Explorer
                                             sub = f"Information of Replacing Order on S & T Bookshop"
   258
   259
                                             print(" Successfully requested for replacement of your order.")
                                             Send_email(data2[8], sub, msq) # Calling Function to send email
   260
```

```
1: Project
     orders.py X
                                            Jenu_email(ualal[J], Job, mag) # outling ronolion to Jenu email
                                            # Real Email is sent here.
  261
                                            # Email is sent actually to the mailbox if the given email really exists
  262
                                            cursor.execute(
  263
                                                f"INSERT INTO orders "
  264
                                                f"(customer_name, address, district, city, state_ut, pincode, country, e_mail, phone_no, pieces, bookname, "
  265
                                                f"user_name, order_type) VALUES ('{data2[1]}', '{data2[2]}', '{data2[3]}', '{data2[4]}', '{data2[5]}', {data2[6]}, "
  266
                                                f"'{data2[7]}', '{data2[8]}', {data2[9]}, {data2[10]}, "
  267
                                                f"'{data2[11]}', '{username}', 'replace')")
  268
                                            cam.commit()
  269
                                            cursor.execute(f"DELETE FROM orders WHERE ord_no = {ID}")
  270
                                            cam.commit()
  271
                                            print(
  272
                                                    Our courier boy will come within the next day to collect the package after verifying its condition, '
  273
                                                    be ready with the packed book')
  274
                                            print(
  275
                                                    Replacement will be completed within 7 working days until your order has been delivered. ")
  276
                                        elif confirm in ['n', 'no']:
  277
                                            print("Replacement Cancelled")
  278
                                            print("Exiting")
  279
                                            return
   280
  281
                                    else:
  282
                                        print('Invalid option choosen.')
                            elif c in ['n', 'no']:
  283
                                print("Exiting")
  284
                                return
                            else:
                                print("Wrong Input")
287
                                again_t = str(input("Do you want to return or replace any of the above orders? (y/n) ")).lower()
  288
                                if again_t in ['yes', 'y']:
  289
                                    work(again_t)
  290
```

```
1: Project
      orders.py X
                                     WOLK (again_ t)
                                elif again_t in ['n', 'no']:
  291
                                     print("Exiting")
  292
  293
                                     return
                                else:
  294
                                    print("Invalid Input.")
  295
  296
                        work(choice)
  297
               cam.close()
  298
               return
  299
  300
  301
  302
                                           Below Functions are for Administrator
  303
  304
  305
  306
            '''This function will allow the administrator to cancel as many orders he wants by specifying the Order ID
  307
              An email will be sent to the user with details. Order can be cancelled only within 7 days after placing order'''
  308
  309
           def Cancel_ord(): # Admin can cancel any user's order
  310
  311
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
副
  312
               cursor = cam.cursor(buffered=True)
               cursor.execute("SELECT ord_no FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7")</pre>
  313
               rec = cursor.fetchall()
  314
               if len(rec) == 0:
  315
                    print("No orders to cancel.")
  316
                    print("Sorry")
  317
               else:
  318
Explorer
                    print(f"Only {len(rec)} order(s) is/are eligible to cancel.")
  319
  320
                    num = int(input(f"Enter number of orders you want to cancel (not more than {len(rec)}): "))
                    if num <= len(rec):</pre>
  321
```

```
Project
      orders.py X
                   if num <= len(rec):</pre>
for i in range(num):
   322
                            order_id = int(input(' Enter the Order ID: '))
   323
                            cursor.execute(f"SELECT ord_no FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7 AND ord_no = {order_id}")</pre>
   324
   325
                            data3 = cursor.fetchone()
                            if len(data3) == 1:
   326
                                cursor.execute(f"SELECT * FROM orders WHERE ord_no = {order_id}")
   327
                                data = cursor.fetchone()
   328
                                if data == None:
   329
                                    print(f"Order ID {order_id} does not exist.")
   330
   331
                                else:
                                    cursor.execute("SET SQL_SAFE_UPDATES = 0")
   332
   333
                                    cursor.execute(
                                        f"UPDATE books SET stock_quantity = stock_quantity + {data[10]} WHERE title = '{data[11]}'")
   334
                                    cam.commit()
   335
                                    cursor.execute(f"SELECT publication, author_fname, author_lname FROM books WHERE title = '{data[11]}'")
   336
                                    data2 = cursor.fetchone()
   337
                                    msg = f"Your Order has been cancelled as per your request. Please review the product on our website. \n" \
   338
                                          f"Book Name: {capwords(data[11])} \n" \
   339
                                          f"Publication: {capwords(data2[0])} \n" \
  340
                                          f"Author's Name: {capwords(data2[1])} {capwords(data2[2])} \n" \
   341
                                          f"Order ID: {data[0]} \n" \
   342
                                          f"Name: {capwords(data[1])} \n" \
343
                                          f"Address: {data[2]} \n" \
   344
   345
                                          f"District: {data[3]} \n" \
                                          f"City: {data[4]} \n" \
   346
ii 347
                                          f"State: {data[5]} \n" \
                                          f"Pincode: {data[6]} \n" \
  348
                                          f"Country: {data[7]} \n" \
  349
AWS Explorer
   350
                                          f"Phone: {data[9]} \n" \
                                          f"Pieces Ordered: {data[10]} \n" \
   351
                                          f"Order Date & Time: {data[12]}"
   352
```

```
1: Project
      orders.py X
   351
                                           T"Pleces urgereg: {gata[lu]} \n" \
                                           f"Order Date & Time: {data[12]}"
  352
                                    email = data[8]
   353
                                    sub = 'Regarding Cancellation of Order on S & T Bookshop'
   354
                                    Send_email(email, sub, msg)
   355
                                    cursor.execute(f"DELETE FROM orders WHERE ord_no = {order_id}")
   356
   357
                                     cam.commit()
                                    print("Successfully cancelled your order.")
   358
                            else:
   359
                                print("Order not eligible to cancel.")
   360
                    else:
   361
                        print(f"Not possible to cancel more than {len(rec)} orders.")
   362
                        Cancel_ord()
   363
               cam.close()
   364
                return
   365
   366
   367
            '''This function will allow administrator to update the order address of multiple orders'''
   368
   369
           Ddef Update_ord_addr(): # Admin can update any user's order
   370
   371
                cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
               cursor = cam.cursor(buffered=True)
   372
               cursor.execute("SELECT * FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7")</pre>
   373
               lim = cursor.fetchall()
   374
               if not lim:
  375
                    print("No orders are eligible for update of address.")
   376
   377
                else:
* 378
                    print(f"Only {len(lim)} order(s) is/are eligible for update of address.")
   379
Explorer
                    def work(data):
   380
                        num = int(input('Enter the number of orders\'s addresses you want to update: '))
   381
   382
                        lst = [(data[i][0]) for i in range(len(data))]
```

```
1: Project
     orders.py X
                       if num <= len(data):</pre>
  384
                           for i in range(num):
                               order_id = int(input('Enter the Order ID to update the order: '))
  385
                               if order_id in lst:
  386
                                   new_addr = str(input('
                                                               Enter the new address for order ID ' + str(order_id) + ': '))
  387
                                   new_city = str(input('
                                                               Enter the new city for order ID ' + str(order_id) + ': '))
  388
                                   new_pinc = int(input('
                                                               Enter the new pincode for order ID ' + str(order_id) + ': '))
  389
                                                               Enter the new district for Order ID' + str(order_id) + ': '))
                                   new_dist = str(input('
  390
                                                               Enter the new State/UT for Order ID' + str(order_id) + ': '))
                                   new_stat = str(input('
  391
                                    cursor.execute("SET SQL_SAFE_UPDATES = 0")
  392
                                   cursor.execute("UPDATE orders SET address = '{}' WHERE ord_no = {}".format(new_addr, order_id))
  393
  394
                                    cam.commit()
                                    cursor.execute("UPDATE orders SET city = '{}' WHERE ord_no = {}".format(new_city, order_id))
  395
                                    cam.commit()
  396
                                   cursor.execute("UPDATE orders SET pincode = {} WHERE ord_no = {}".format(new_pinc, order_id))
  397
                                    cam.commit()
  398
                                    cursor.execute("UPDATE orders SET district = '{}' WHERE ord_no = {}".format(new_dist, order_id))
  399
                                    cam.commit()
  400
                                   cursor.execute("UPDATE orders SET state_ut = '{}' WHERE ord_no = {}".format(new_stat, order_id))
  401
                                    cam.commit()
  402
Structure
                                   cursor.execute(f"SELECT e_mail FROM orders WHERE ord_no = {order_id}")
  403
  404
                                    email = cursor.fetchone()
                                   msg = f"Your Order has been updated successfully \n. " \
  405
                                          f"New Location of the order is {new_addr}, {new_city}, {new_pinc}, {new_city}, {new_stat} :)"
  406
  407
                                    sub = "Information of Order Update on S & T Book Shop Management System"
  408
                                   Send_email(email[0], sub, msg)
                                   print('Successfully Updated')
  410
                                   print(f"regarding order update for new location on email address {email[0]}")
                               else:
  411
  412
                                   print("This order ID is not eligible for update of order address.")
  413
                                    continue
                       else:
  414
```

```
1: Project
     orders.py X
                                    continue
  414
                       else:
                            print("Invalid Input. Try Again")
  415
                           work(data)
  416
                   work(lim)
  417
               cam.close()
  418
               return
  419
  420
  421
           '''This function will allow administrator to view all orders in database of every user.'''
  422
  423
           def View_all_ords(): # Admin can view all user's order
  424
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
  425
               cursor = cam.cursor(buffered=True)
  426
               print(" 1 -> View all New Orders Placed \n"
  427
                     " 2 -> View all Orders for Return \n"
  428
                     " 3 -> View all Orders for Replacement \n"
  429
                     " 4 -> View all Orders including all categories")
  430
               ch = int(input(" Enter your choice (1 to 4): "))
  431
  432
               def start(choice):
  433
                   if choice == 1:
  434
  435
                       cursor.execute("SELECT * FROM orders WHERE order_type = 'new' ORDER BY ord_no DESC")
                       data1 = cursor.fetchall()
  436
                       if not data1:
  437
                            print("No orders.")
  438
  439
                       else:
                           for row in data1:
* 440
                                tp = row
  441
Explorer
  442
                                print("
                                           Order ID
                                                        : ", tp[0])
                                                         : ", capwords(tp[11]))
  443
                                print("
                                           Book Name
                                print("
                                                           : ", tp[10])
  444
                                           Pieces
```

```
1: Project
      orders.py X
                                 print("
                                             Customer Name : ", capwords(tp[1]))
   446
                                 print("
                                             Address
                                                             : ", tp[2])
                                                            : ", tp[3])
                                 print("
                                             District
   447
                                                             : ", tp[4])
                                 print("
                                             City
   448
                                                             : ", tp[5])
                                 print("
                                             State
   449
                                                             : ", tp[6])
                                 print("
                                             Pincode
   450
                                 print("
                                             Country
                                                             : ", tp[7])
   451
                                 print("
                                             E-mail Address : ", tp[8])
   452
                                 print("
                                                            : ", tp[9])
   453
                                             Phone
                                 print("
                                             Order Date
                                                           : ", tp[12])
   454
                                                            : ", tp[13])
                                 print("
                                             Username
   455
                                 print()
   456
                    elif choice == 2:
   457
                         cursor.execute("SELECT * FROM orders WHERE order_type = 'return' ORDER BY ord_no DESC")
   458
                        data2 = cursor.fetchall()
   459
                        if not data2:
   460
                             print("No orders.")
   461
   462
                         else:
                             for row in data2:
   463
                                 tp = row
   464
Structure
   465
                                 print("
                                             Order ID
                                                          : ", tp[0])
                                 print("
                                             Book Name
                                                            : ", capwords(tp[11]))
   466
έ¥.
                                                             : ", tp[10])
                                 print("
                                             Pieces
<del>-</del> 467
                                 print("
                                             Customer Name : ", capwords(tp[1]))
   468
                                                            : ", tp[2])
   469
                                 print("
                                             Address
   470
                                 print("
                                             District
                                                            : ", tp[3])
ii 471
                                 print("
                                             City
                                                             : ", tp[4])
<del>*</del> 472
                                                             : ", tp[5])
                                 print("
                                             State
                                                            : ", tp[6])
                                 print("
                                             Pincode
AWS Explorer
   474
                                 print("
                                             Country
                                                             : ", tp[7])
                                             E-mail Address : ", tp[8])
   475
                                 print("
                                 print("
                                                             : ", tp[9])
   476
                                             Phone
```

```
Project
      orders.py X
  477
                                print("
                                           Order Date : ", tp[12])
                                print("
                                                          : ", tp[13])
                                           Username
  478
                                print()
  479
                   elif choice == 3:
  480
                        cursor.execute("SELECT * FROM orders WHERE order_type = 'replace' ORDER BY ord_no DESC")
  481
                        data1 = cursor.fetchall()
  482
                       if not data1:
  483
                            print("No orders.")
  484
                        else:
  485
                            for row in data1:
  486
                                tp = row
  487
                                           Order ID
                                                         : ", tp[0])
  488
                                print("
                                print("
                                           Book Name
                                                          : ", capwords(tp[11]))
  489
                                                           : ", tp[10])
                                print("
                                           Pieces
  490
                                           Customer Name : ", capwords(tp[1]))
                                print("
  491
                                print("
                                           Address
                                                          : ", tp[2])
  492
                                                          : ", tp[3])
                                print("
                                           District
  493
                                                           : ", tp[4])
                                print("
  494
                                           City
                                print("
                                           State
                                                           : ", tp[5])
  495
                                print("
                                                          : ", tp[6])
  496
                                           Pincode
                                print("
                                                          : ", tp[7])
  497
                                           Country
                                print("
                                           E-mail Address : ", tp[8])
  498
61
  499
                                                           : ", tp[9])
                                print("
                                           Phone
                                                         : ", tp[12])
  500
                                print("
                                           Order Date
Favorites
                                print("
                                                           : ", tp[13])
  501
                                           Username
  502
                                print()
ċί
  503
                   elif choice == 4:
                        cursor.execute("SELECT * FROM orders ORDER BY ord_no DESC")
  504
                       data = cursor.fetchall()
  505
AWS Explorer
  506
                        if not data:
                            print("No orders.")
  507
                        else:
  508
```

```
1: Project
      orders.py X
                            for row in data:
   509
                                tp = row
   510
                                print("
                                            Order ID
                                                           : ", tp[0])
   511
                                                          : ", capwords(tp[11]))
                                print("
                                            Book Name
   512
                                print("
                                           Pieces
                                                           : ", tp[10])
   513
                                           Customer Name : ", capwords(tp[1]))
                                print("
   514
                                                           : ", tp[2])
                                print("
   515
                                            Address
                                print("
                                           District
                                                           : ", tp[3])
   516
                                print("
                                            City
                                                           : ", tp[4])
   517
                                print("
                                           State
                                                           : ", tp[5])
   518
                                print("
                                            Pincode
                                                           : ", tp[6])
   519
                                                           : ", tp[7])
                                print("
                                            Country
   520
                                print("
                                            E-mail Address : ", tp[8])
   521
                                                           : ", tp[9])
                                print("
                                            Phone
   522
                                                           : ", tp[12])
                                print("
   523
                                            Order Date
                                print("
                                           Username
                                                           : ", tp[13])
   524
                                                          : ", tp[14])
                                print("
                                           Order Type
   525
   526
                                if tp[14] not in ['return', 'replace']:
                                     cursor.execute(
   527
                                        f"SELECT DATE(order_dt) FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7 AND ord_no = {tp[0]}")
   528
   529
                                     data2 = cursor.fetchone()
FI 530
                                    if not data2:
   531
                                         cursor.execute(
                                             f"SELECT DATE(DATE_ADD(order_dt, INTERVAL 7 DAY)) FROM orders WHERE ord_no = {tp[0]}")
   532
   533
                                         data3 = cursor.fetchone()
   534
                                         print(f"
                                                     Order Status
                                                                   : Delivered on {data3[0]}")
  535
                                     else:
                                         print("
                                                    Order Status : In transit")
   536
Explorer
   537
                                print()
   538
                    else:
§ 539
                        ch2 = int(input("Wrong Input. Enter Again: "))
```

```
Project
      orders.py X
÷
                        ch2 = int(input("Wrong Input. Enter Again: "))
   539
                        start(ch2)
   540
   541
               start(ch)
   542
               cam.close()
   543
               return
   544
   545
   546
            '''This function will allow administrator to view all orders of a user'''
   547
   548
           Odef View_all_u(): # Admin can view any user's order particularly
   549
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
   550
               cursor = cam.cursor(buffered=True)
   551
               cursor.execute("SELECT ord_no FROM orders")
   552
               data = cursor.fetchall()
   553
               if not data:
   554
                    print("No orders in database.")
   555
                else:
   556
                    user = str(input("Enter the customer's username: "))
   557
                    cursor.execute(f"SELECT * FROM orders WHERE user_name = '{user}' ORDER BY order_dt DESC")
   558
                   data2 = cursor.fetchall()
   559
560
                   if data2 == None:
                        print(f"No orders of username '{user}'.")
   561
  562
                    else:
2: Favorites
   563
                        for row in data2:
   564
                            tp = row
                                                    : ", tp[0])
                            print("
                                       Order ID
  565
                                                    : ", capwords(tp[11]))
                            print("
                                       Book Name
   566
WS Explorer
                                                      : ", tp[10])
   567
                            print("
                                       Pieces
   568
                            print("
                                       Customer Name : ", capwords(tp[1]))
                            print("
                                                       : ", tp[2])
  569
                                        Address
```

```
orders.py X
                            print("
                                               : ", tp[2])
<del>~</del>1 569
                                       Address
  570
                           print("
                                       District
                                                      : ", tp[3])
                           print("
                                       City
                                                      : ", tp[4])
   571
                           print("
                                                      : ", tp[5])
                                       State
   572
                           print("
                                       Pincode
                                                      : ", tp[6])
   573
                                                      : ", tp[7])
  574
                            print("
                                       Country
                           print("
                                       E-mail Address : ", tp[8])
   575
                                                      : ", tp[9])
  576
                           print("
                                       Phone
                           print("
                                                    : ", tp[12])
                                       Order Date
   577
                                      Order Type : ", capwords(tp[14]))
                           print("
   578
                           if tp[14] not in ['return', 'replace']:
   579
   580
                                cursor.execute(
                                    f"SELECT DATE(order_dt) FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7 AND ord_no = {tp[0]}")
   581
                                data2 = cursor.fetchone()
   582
                                if not data2:
   583
  584
                                    cursor.execute(
                                        f"SELECT DATE(DATE_ADD(order_dt, INTERVAL 7 DAY)) FROM orders WHERE ord_no = {tp[0]}")
  585
                                    data3 = cursor.fetchone()
   586
                                                Order Status : Delivered on {data3[0]}")
   587
                                    print(f"
                                else:
  588
                                               Order Status : In transit")
   589
                                    print("
                           print()
   590
               cam.close()
  591
   592
               return
  593
  594
ii 595
           '''This function will allow administrator to view all orders of a particular date'''
  596
           Ddef View_ord_dt(): # Admin can view any user's order by date
  597
AWS Explorer
  598
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
  599
               cursor = cam.cursor(buffered=True)
               cursor.execute("SELECT ord_no FROM orders")
  600
```

```
Project
     orders.py X
÷
               data = cursor.fetchall()
  601
               if not data:
  602
                   print("No orders in database.")
  603
               else:
  604
                   dt = str(input("Enter date (YYYY-MM-DD): "))
  605
                   cursor.execute(f"SELECT * FROM orders WHERE DATE(order_dt) = '{dt}'")
  606
                   data = cursor.fetchall()
  607
                   for row in data:
  608
                        tp = row
  609
                                   Order ID
                                                 : ", tp[0])
                        print("
  610
                        print("
                                   Book Name
                                                  : ", capwords(tp[11]))
  611
                                                   : ", tp[10])
                        print("
                                   Pieces
  612
                                   Customer Name : ", capwords(tp[1]))
                        print("
  613
                        print("
                                   Address
                                                   : ", tp[2])
  614
                                   District
                                                  : ", tp[3])
                        print("
  615
                        print("
                                   City
                                                   : ", tp[4])
  616
                                                   : ", tp[5])
  617
                        print("
                                   State
                        print("
                                   Pincode
                                                   : ", tp[6])
  618
                        print("
                                                   : ", tp[7])
                                   Country
  619
                        print("
                                   E-mail Address : ", tp[8])
  620
  621
                        print("
                                                   : ", tp[9])
                                   Phone
                        print("
                                   Order Date
                                                   : ", tp[12])
  622
                        print("
                                                  : ", tp[13])
  623
                                   Username
                        print("
                                   Order Type
                                                : ", capwords(tp[14]))
  624
Favorites
  625
                        if tp[14] not in ['return', 'replace']:
  626
                            cursor.execute(f"SELECT DATE(order_dt) FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7 AND ord_no = {tp[0]}")</pre>
ċί
                            data2 = cursor.fetchone()
                            if not data2:
  628
                                cursor.execute(f"SELECT DATE(DATE_ADD(order_dt, INTERVAL 7 DAY)) FROM orders WHERE ord_no = {tp[0]}")
  629
AWS Explorer
  630
                                data3 = cursor.fetchone()
                                            Order Status : Delivered on {data3[0]}")
                                print(f"
  631
```

else:

632

```
1: Project
      orders.py X
                                print("
                                           Order Status : In transit")
  633
                        print()
  634
               cam.close()
  635
  636
               return
  637
  638
                                                    Below Functions are only for non-admin users
  639
  640
  641
  642
           '''This function will allow users to cancel their order only if they cancel it within 7 days of the date since
  643
              order is placed. An email will be sent to the user with the details.'''
  644
  645
           def Cancel_ord_u(username):
  646
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
  647
               cursor = cam.cursor(buffered=True)
  648
               cursor.execute(
  649
                   f"SELECT * from orders where user_name = '{username}' and DATEDIFF(NOW(), order_dt) <= 7 ORDER by ord_no DESC")
  650
               data = cursor.fetchall()
  651
               list1 = []
  652
               if not data:
  653
                   print("None of your orders are eligible to cancel. Sorry :)")
               else:
                   print("This is the list of your orders which are eligible to cancel: ")
  656
2: Favorites
  657
                   for row in data:
  658
                        tp = row
                       list1.append(tp[0])
  659
                                                : ", tp[0])
                        print("
                                   Order ID
  660
Explorer
  661
                        print("
                                   Book Name
                                               : ", capwords(tp[11]))
  662
                                                  : ", tp[10])
                        print("
                                   Pieces
                                   Customer Name : ", capwords(tp[1]))
  663
                        print("
```

```
Project
      orders.py X
<del>::</del>|
                                    Customer Name : ", capwords(tp[1]))
  663
                        print("
                        print("
                                                    : ", tp[2])
                                    Address
  664
                        print("
                                                    : ", tp[3])
                                    District
  665
                        print("
                                    City
                                                    : ", tp[4])
  666
                                                    : ", tp[5])
                        print("
                                    State
  667
                                                    : ", tp[6])
  668
                        print("
                                    Pincode
                        print("
                                    Country
                                                    : ", tp[7])
  669
                                    E-mail Address : ", tp[8])
  670
                        print("
                                                    : ", tp[9])
  671
                        print("
                                    Phone
                        print("
                                    Order Date
                                                    : ", tp[12])
  672
                                    Order Type
                                                    : ", tp[14])
  673
                        print("
  674
                        if tp[14] not in ['return', 'replace']:
  675
                            cursor.execute(f"SELECT DATE(order_dt) FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7 AND ord_no = {tp[0]}")</pre>
  676
                            data2 = cursor.fetchone()
  677
                            if not data2:
  678
                                 cursor.execute(f"SELECT DATE(DATE_ADD(order_dt, INTERVAL 7 DAY)) FROM orders WHERE ord_no = {tp[0]}")
  679
                                 data3 = cursor.fetchone()
  680
                                 print(f"
                                             Order Status : Delivered on {data3[0]}")
  681
  682
                            else:
                                            Order Status : In transit")
  683
                                 print("
  684
                        print()
  685
                    def choose():
  686
Favorites
                        choice = int(input("Enter the Order ID to cancel: "))
  687
  688
                        return choice
ċίΙ
  689
                    ch = choose()
  690
                    if ch in list1:
  691
AWS Explorer
  692
                        cursor.execute(f"SELECT * FROM orders WHERE ord_no = {ch}")
  693
                        data2 = cursor.fetchone()
                        cursor.execute(f"SELECT publication, author_fname, author_lname FROM books WHERE title = '{data2[11]}'")
```

```
Project
     orders.py X
÷
                        data3 = cursor.fetchone()
  695
  696
                              f"Book Name: {capwords(data2[11])} \n" \
  697
                              f"Publication: {capwords(data3[0])} \n" \
  698
  699
                              f"Order ID: {data2[0]} \n" \
  700
                              f"Name: {capwords(data2[1])} \n" \
  701
                              f"Address: {data2[2]} \n" \
  702
                              f"District: {data2[3]} \n" \
  703
                              f"City: {data2[4]} \n" \
  704
                              f"State: {data2[5]} \n" \
  705
                              f"Pincode: {data2[6]} \n" \
  706
                              f"Country: {data2[7]} \n" \
  707
                              f"Phone: {data2[9]} \n" \
  708
                              f"Pieces Ordered: {data2[10]} \n" \
  709
                              f"Order Date & Time: {data2[12]} \n" \
  710
  711
                       email = data2[8]
  712
  713
                       Send_email(email, sub, msg)
  714
                       cursor.execute("SET SQL_SAFE_UPDATES = 0")
  715
  716
                        cursor.execute(
  717
                        cam.commit()
  718
  719
                       cam.commit()
  720
                       print("Successfully Cancelled the Order.")
  721
                   if len(list1) > 1:
  722
  723
                       if ch2 == 'y' or ch2 == 'Y':
  724
                            Cancel_ord_u(username)
  725
```

```
PODETOGETOR, GOTTOL TRAME, GOTTOL TRAME I NOT BOOKS MILKE
msq = f"Your Order has been cancelled as per your request. Please review the product on our website. \n\n" \
      f"Author's Name: {capwords(data3[1])} {capwords(data3[2])} \n" \
      f"Order Status: Cancelled on {datetime.datetime.now()}"
sub = 'Regarding Cancellation of Order on S & T Bookshop'
    f"UPDATE books SET stock_quantity = stock_quantity + {data2[10]} WHERE title = '{data2[11]}'")
cursor.execute(f"DELETE FROM orders where ord_no = {ch}")
ch2 = str(input(" Want to cancel more Orders? (y OR n) "))
```

```
Project
      orders.py X
                            Cancel_ord_u(username)
  725
                        elif ch2 == 'n' or ch2 == 'N':
  726
                            cam.close()
  727
                            return
  728
                        else:
  729
                            print("Wrong Input")
  730
                            return
  731
                    else:
  732
  733
                        pass
  734
  735
           '''This function will allow administrator to update his/her order address if they do it within 7 days after
  736
              order is being placed. An email will be sent with few order details and new location of delivery.'''
  737
  738
           def Up_ord_au(username):
  739
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
  740
               cursor = cam.cursor(buffered=True)
  741
               cursor.execute(f"SELECT * FROM orders WHERE user_name = '{username}' and DATEDIFF(NOW(), order_dt) <= 7 ORDER BY ord_no DESC")</pre>
  742
               data = cursor.fetchall()
  743
               list1 = []
  744
  745
               if data == None:
  746
                    print("No 'IN TRANSIT' orders in database for your username.")
  747
               else:
                    print("This is the list of your orders eligible for updating the address: ")
  748
Favorites
                   for row in data:
  749
  750
                        tp = row
άį
                        list1.append(tp[0])
                                                    ", tp[0])
  752
                        print("
                                   Order ID:
                                                    ", capwords(tp[11]))
                                   Book Name:
  753
                        print("
AWS Explorer
  754
                        print("
                                                    ", tp[10])
                                   Pieces:
                                   Customer Name: ", capwords(tp[1]))
                        print("
  755
                        print("
                                                    ", tp[2])
  756
                                   Address:
```

```
Project
      orders.py X
<del>~</del>i 757
                        print("
                                                    ", tp[3])
                                   District:
  758
                        print("
                                   City:
                                                    ", tp[4])
                                   State/UT:
                                                    ", tp[5])
                        print("
  759
                                                    ", tp[6])
                                   Pincode:
  760
                        print("
                        print("
                                   Country:
                                                    ", tp[7])
  761
                        print("
                                   E-mail Address: ", tp[8])
  762
                                                    ", tp[9])
  763
                        print("
                                    Phone:
                        print("
                                   Order Date:
                                                    ", tp[12])
  764
                        print()
  765
  766
                    def choose():
  767
                        choice = int(input("Enter the Order ID to update address: "))
  768
                        return choice
  769
                    ch = choose()
  770
                    if ch in list1:
  771
                        new_addr = str(input("Enter the new address: "))
  772
                        new_city = str(input("Enter City: "))
  773
                        new_pinc = int(input('Enter pincode: '))
  774
  775
                        new_dist = str(input("Enter District: "))
                        new_stat = str(input("Enter State/UT" + ': '))
  776
  777
                        cursor.execute("SET SQL_SAFE_UPDATES = 0")
                        cursor.execute(f"UPDATE orders SET address = '{new_addr}' WHERE ord_no = {ch}")
έű
                        cam.commit()
  779
                        cursor.execute(f"UPDATE orders SET city = '{new_city}' WHERE ord_no = '{ch}'")
  780
  781
                        cam.commit()
                        cursor.execute(f"UPDATE orders SET pincode = {new_pinc} WHERE ord_no = {ch}")
  782
NI 783
                        cam.commit()
                        cursor.execute(f"UPDATE orders SET district = '{new_dist}' WHERE ord_no = '{ch}'")
  784
                        cam.commit()
  785
AWS Explorer
  786
                        cursor.execute(f"UPDATE orders SET state_ut = '{new_stat}' WHERE ord_no = '{ch}'")
  787
                        cam.commit()
                        print('Successfully Updated')
  788
```

```
Project
      orders.py ×
                        cursor.execute(f"SELECT email FROM accounts WHERE username = '{username}'")
  789
                        email = cursor.fetchone()
  790
                        msg = f"Your Order has been updated successfully. \n" \
  791
                              f"New Location of the order is {new_addr}, {new_city}, {new_pinc}, {new_city}, {new_stat} :)"
  792
                        sub = "Information of Order Update on S & T Book Shop Management System"
  793
                        Send_email(email[0], sub, msg)
  794
                        print()
  795
  796
                   if len(list1) > 1:
  797
                        ch2 = str(input(" Want to update more Orders? (y OR n): "))
  798
                       if ch2 == 'y' or ch2 == 'Y':
  799
                            Up_ord_au(username)
  800
                        elif ch2 == 'n' or ch2 == 'N':
  801
                            return
  802
  803
                        else:
                            print("Wrong Input")
  804
                            return
  805
                   else:
  806
  807
                        pass
               cam.close()
  808
  809
               return
  810
  811
           '''This function will allow the signed in user to view all their details of every order placed since
  812
Favorites
              registered with detail of order type (i.e, new or for replace or for return)'''
  813
  814
           def View_au(username):
  815
               cam = mycam.connect(host='localhost', user='root', passwd='Rinshu@03', database='book_shop')
  816
  817
               cursor = cam.cursor(buffered=True)
AWS Explorer
  818
               print("This is the list of your orders.: ")
               cursor.execute(f"SELECT * FROM orders WHERE user_name = '{username}' ORDER BY ord_no DESC")
  819
  820
               data = cursor.fetchall()
```

```
1: Project
      orders.py X
               if data == None:
  821
                    print("No orders in database.")
  822
  823
               else:
                    print("
                                               Latest Orders Sorted by Date:. ")
  824
                    for row in data:
  825
                        tp = row
  826
                        print("
                                   Order ID
                                                 : ", tp[0])
  827
                        print("
                                   Book Name
                                                  : ", capwords(tp[11]))
  828
                                                   : ", tp[10])
                        print("
                                   Pieces
  829
                                   Customer Name : ", capwords(tp[1]))
                        print("
  830
                                   Address
                                                   : ", tp[2])
  831
                        print("
                                   District
                                                   : ", tp[3])
                        print("
  832
                                                   : ", tp[4])
                        print("
                                   City
  833
                                                   : ", tp[5])
                        print("
                                   State/UT
  834
                        print("
                                   Pincode
                                                   : ", tp[6])
  835
                        print("
                                   Country
                                                   : ", tp[7])
  836
                        print("
                                   E-mail Address : ", tp[8])
  837
                                                   : ", tp[9])
                        print("
                                   Phone
  838
                                                   : ", tp[12])
                        print("
                                   Order Date
  839
                        print("
                                   Order Type
                                                  : ", capwords(tp[14]))
  840
  841
ŔΪ
                        if tp[14] not in ['return', 'replace']:
  842
                            cursor.execute(f"SELECT DATE(order_dt) FROM orders WHERE DATEDIFF(NOW(), order_dt) <= 7 AND ord_no = {tp[0]}")</pre>
  843
  844
                            data2 = cursor.fetchone()
2: Favorites
  845
                            if not data2:
                                cursor.execute(f"SELECT DATE(DATE_ADD(order_dt, INTERVAL 7 DAY)) FROM orders WHERE ord_no = {tp[0]}")
  846
                                data3 = cursor.fetchone()
  847
                                print(f"
                                             Order Status : Delivered on {data3[0]}")
  848
Explorer
  849
                            else:
                                            Order Status : In transit")
                                print("
  850
  851
                        print()
```

ucti	852	cam.close()		
7: Stu	852 853	cam.close() return		
	854			
	855	<b>p#</b>		
rites	856 857	#	End of Module	1
avo	857	<u></u>		
;i	858			