Auction Bidding Management System

Version 1.0 (2020-2021)

Computer Science (083) Project

Developed By

KRISH CHUGH

ARNAV AGARWAL

Delhi Public School, R.K.Puram, New Delhi

www.dpsrkp.net

Index

Certificate

This is to certify that the **Auction Bidding System 1.0** Computer Science project is developed by **Arnav Agarwal** and **Krish Chugh** under my supervision in the session 2020-2021.

The work done by them is original.

Ms. Hema Jain
Computer Science Teacher
Date: 20/11/2020

Acknowledgement

I would like to express my sincere gratitude to my computer teacher Ms. Hema Jain for her vital support, guidance and encouragement without which this project would not come forth from my side. She helped me complete the project by giving ideas, thoughts and made this project easy and accurate.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way, without which I would be unable to complete my project.

REFERENCES:

1) CLASSNOTES
2) COMPUTER SCIENCE WITH PYTHON BY SUMITA ARORA

Introduction

In today's world of growing requirements of effective and efficient solutions for different problems in various sectors, with our knowledge of Python and SQL, in this project, we have tried to provide a simple solution for the Auction Bidding System.

Our project has the complete management solutions to take care of all the sales events wherein all the potential buyers can place competitive bids on assets. This would help in providing both the buyers and the sellers a good deal for the assets to be exchanged.

The program has been designed for buying and selling goods in a variety of categories, most prominently antiques and artworks, but also furniture and electronics. When the bidding process is initiated for an asset, various potential buyers are asked to put their bids and the process continues until there is anybody left to contest the latest bidding amount. The product is then sold to the one with the highest bid.

The entire program has been divided into small user defined functions to take care of all kinds of functionalities provided. The following is the list of all the Functions along with their description.

Sno	Function Name	Description
1	Add_Bidder()	1. To add a bidder in the list of bidders. 2. To ask the new bidder to input his details. 3. To check if the bidder is underage or not. 4. Create a table dedicated to the bidder.
2	Upload_Item()	To add an Item into the list of items available for auction.
3	Search_Item(a)	To search for an item in the list of items based on ID or Name of item controlled via parameter 'a'
4	ViewBidder(a)	1. To search for a bidder in the list of bidders and view details of that bidder. 2. To view items purchased by a bidder during auctions.
5	View(status)	To view items matching a certain criteria such as category or status.
6	Bidding()	1. To initiate the process of bidding 2. To update table ITEMS and change the status of items which have been sold 3. To update the table BIDDERS with the item that has been purchased.

7	Generate_Bill()	1. To generate bill for bidder whose username is entered 2. To input mode of payment from bidder 3. To clear all dues of the bidder
8	Remove_from_Cart()	1. To help the bidder remove an item from their cart. 2. To add the item back into list of items available for auction 3. To update the bill amount for the bidder.

Tables with structure used in the program

File/Table Name: BIDDERS

Sno	FieldName/ Column Name	Data Type	Description
1	ID	INTEGER(4)	Assigned randomly.Unique.
2	NAME	CHAR(24)	Name of Bidder.(not null)
3	AGE	INTEGER(4)	Has to be more than 18.
4	USERNAME	VARCHAR(25)	Used to identify the bidder.Primary Key.
5	PASSWORD	VARCHAR(10)	For security purposes.(not null)
6	QTY	INTEGER(6)	Quantity of items purchased by a bidder.
7	TOTAL_AMOUNT	INTEGER(10)	Total bill due to be paid.
8	DOP	DATE	Date of purchase of last item.
9	EMAIL	VARCHAR(40)	Email of bidder.
10	MOBILE	VARCHAR(11)	Mobile number of bidder.

File/Table Name: ITEMS

Sno	FieldName/ Column Name	Data Type	Description
1	ID	INTEGER(4)	Primary Key to identify each ITEM.Assigned by auctioneer.
2	NAME	CHAR(24)	Name of item(not null)
3	CATEGORY	CHAR(20)	Category to which item belongs.For eg.Artwork,Antiques,Furniture,Electronics
4	BASE_PRICE	INTEGER(10)	Minimum price at which item can be sold,assigned by the auctioneer.(not null)
5	BID_AMOUNT	INTEGER(10)	The price at which item is being sold.
6	STATUS	CHAR(5)	Whether the item has been sold or not.Can be either ‘open’ or ‘close’.
7	DOP	DATE	Date when item was purchased
8	BUYER	CHAR(20)	ID of buyer who purchased item

File/Table Name: BIDDERNAME(each bidder has individual table)

Sno	FieldName/ Column Name	Data Type	Description
1	ID	INTEGER(2)	Primary Key to identify each Item purchased by the bidder.
2	NAME	CHAR(24)	Name of item(not null).
3	BASE_PRICE	INTEGER(10)	Base price of item purchased.
4	BID_PRICE	INTEGER(10)	Price at which item was purchased.
5	CATEGORY	CHAR(20)	Category to which item belongs.For eg.Artwork,Antiques,Furniture,Electronics
6	DOP	DATE	Date of purchase of item.
7	BILL_PAID	CHAR(5)	Whether the bill has been paid or not.

Salient Features of the Project

- User friendly menu driven options

  User specified parameter in VIEW() to ask if the user wants to view items open for bidding or closed for bidding.

- Exceptions used to avoid inconvenience to users.Examples:
  1. The case if the database already exists.
  2. The case if the table already exists.
  3. If the username entered in BIDDING() does not exist.
- Data Validation to avoid wrong/ambiguous/invalid data entry.
  1. Each bidder is required to enter their age and only allowed to continue in case their age is 18 years or above.
  2. In the function SearchItem() if the user enters the wrong ID/Name of item an error message “Item not found “ is displayed.
  3. In UploadItem() an ID is only assigned to an item if it has not been taken already.
  4. In Bidding() appropriate message is displayed if wrong ID is entered or item is already sold
- Security:Each bidder is required to enter a password. Every time the bidder wishes to carry out a function such as view his purchased items,participate in bidding or pay the bill, he is required to enter the password. The user is given 3 attempts to enter the password.

```
# Project Title   : Auction Bidding System
# Version           : 1.0 2020-2021
# Developed By  : Krish Chugh and Arnav Agarwal
# Guide              : Ms. Hema Jain
# Last Updated On   : 2021-01-28


import MySQLdb
import random

db=MySQLdb.connect(host="localhost",user="root",password="ArnaV2602")
MyCur=db.cursor()

try:
   MyCur.execute("Create Database Auct")
   MyCur.execute("Use Auct")
except MySQLdb._exceptions.ProgrammingError :
   MyCur.execute("Use Auct")



try:
  MyCur.execute("""
  Create Table BIDDERS(
    ID INTEGER(4) UNIQUE,
    NAME CHAR(24) NOT NULL,
    AGE INTEGER(4) NOT NULL,
    USERNAME VARCHAR(25) PRIMARY KEY,
    PASSWORD VARCHAR(20) NOT NULL,
    QTY INTEGER(6),
    TOTAL_AMOUNT INTEGER(10),
    DOP DATE,
    EMAIL VARCHAR(40),
    MOBILE VARCHAR(11))""")
except MySQLdb.OperationalError:
   print("Table already exists")
```

```
''''''''''''''''''''''''''''''''''''''''''''
To add a bidder to the list
''''''''''''''''''''''''''''''''''''''''''''

def Add_Bidder():
    choice = 'Y'
    Z='True'
    while choice == 'Y' or choice == 'y':
          biddername=input("Enter name of bidder")
          bid_username=input("Enter username of bidder (The username can only be a
combination of alphabets, numbers and underscores without any spaces)")
          S = "SELECT * FROM BIDDERS WHERE USERNAME = '" + bid_username + "'"
          N = MyCur.execute(S)
          if N>0:
             print("Username already taken. Please try again!")
          else:
             age = int(input("Enter age of bidder"))
             if age > 18:
                password=input("Enter password")
                bidder_id = random.randint(100,1000) #check and add validation
                email = input("Enter you email id")
                mobile = input("Enter your mobile number")
                try:
                   MyCur.execute("CREATE    TABLE    "+bid_username+"(ID    INTEGER(4)
PRIMARY KEY,NAME CHAR(24) NOT NULL, BASE_PRICE INTEGER(10), BID_PRICE INTEGER(10),
CATEGORY CHAR(10), DOP DATE, BILL_PAID CHAR(5))")
                   print("Table created")
                   MyCur.execute("INSERT                  INTO                  BIDDERS
VALUES("+str(bidder_id)+",'"+biddername+"',"+str(age)+",'"+bid_username+"','"+passwo
rd+"',0,0,NULL,'"+email+"','"+mobile+"')")
                   print("Bidder added")
                   db.commit()
                except Exception as e:
                   print(str(e))
             else:
                print("Sorry, you are under age for bidding!")
          choice = input("Do you want to add another bidder(Y/N)?")
```

```
''''''''''''''''''''''''''''''''''''''''''''
To upload any item for bidding
''''''''''''''''''''''''''''''''''''''''''''

def Upload_Item():
   try:
       MyCur.execute("Create Table Items(ID INTEGER(4) PRIMARY KEY,NAME CHAR(24) NOT
NULL, CATEGORY CHAR(20), BASE_PRICE INTEGER(10) NOT NULL,BID_AMOUNT  INTEGER(10),
STATUS CHAR(5), DOP DATE, BUYER CHAR(20))")
   except:
       print("Table already exists")
   choice = 'Y'
   while choice == 'Y' or choice == 'y':
        Id=input("Enter the Id")
        N = MyCur.execute("SELECT * FROM Items WHERE ID ="+Id)
        if N>0:
            print("ID already taken")
            break
        else:
           item=input("Enter the name of item")
           category = input ("Enter the item's category")
           base_price=input("Enter base price")
           status= "Open"
           SQL="INSERT                              INTO                              ITEMS
VALUES("+Id+",'"+item+"','"+category+"',"+base_price+",NULL,'"+status+"',NULL,NULL)"
           MyCur.execute(SQL)
           db.commit()
           choice = input("Do you want to upload another item(Y/N)?")
```

```
'''''''''''''''''''''''''''''''''
To search any item
'''''''''''''''''''''''''''''''''

def SearchItem():
    type = input("Do you want to search on item ID or name(ID/NAME)?")
    if type == "ID":
        S=input("Enter Id of item to search for")
        SQL="SELECT * FROM ITEMS WHERE ID="+S+""
    elif type == "NAME":
        S=input("Enter Name of item to search for")
        SQL="SELECT * FROM ITEMS WHERE NAME='"+S+"'"
    N=MyCur.execute(SQL)
    if N>0:
        MyCur.execute(SQL)
        R=MyCur.fetchone()
        for i in R:
            print(i,end = "   |   ")
        print('')
    else:
        print("Result not Found")



''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
To search any bidder and view items purchased by them
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

def ViewBidder():
   type=(input("Do you want to search on bidder ID or name(ID/NAME)?")).upper()
   if type=="ID":
      S=input("Enter Id of bidder to search for")
      SQL="SELECT * FROM BIDDERS WHERE ID="+S+""
   elif type == "NAME":
        S=input("Enter Name of bidder to search for")
        SQL="SELECT * FROM BIDDERS WHERE NAME='"+S+"'"
   N=MyCur.execute(SQL)
   if N>0:
        MyCur.execute(SQL)
        R=MyCur.fetchone()
        bid_username=R[3]
        S="SELECT * FROM "+bid_username+""
        MyCur.execute(S)
        H=MyCur.fetchall()
        for i in R:
            print(i,end = "   |   ")
        print('')
        for j in H:
           print(j)
   else:
        print("Result not Found")
```

```
''''''''''''''''''''''''''''''''''''''''''''
To view any item under a certain category
''''''''''''''''''''''''''''''''''''''''''''

def View(status = 'Open'):
  cat = input("Enter the category of the items that you to view/bid - Artwork,
Antiques, Furniture, Electronics")
  SQL = "SELECT * FROM ITEMS WHERE CATEGORY = '"+cat+"' AND STATUS='"+status+"'"
  try:

    N = MyCur.execute(SQL)
    if N>0:
        MyCur.execute(SQL)
        R=MyCur.fetchall()
        for i in R:
            print (i)
        return True
    else:
        print("No items for the searched criteria")
        return False
  except Exception as e:
     print(str(e))
     print(SQL)
```

```
''''''''''''''''''''''''''''''''''''''''''''
To initiate the process of bidding
''''''''''''''''''''''''''''''''''''''''''''

def Bidding():
    A = View('Open')
    if A is True:
        I=input("Enter ID of item to bid for")
        SQL="SELECT * FROM ITEMS WHERE ID="+str(I)+" AND STATUS = 'Open'"
        N=MyCur.execute(SQL)
        if N>0:
            R=MyCur.fetchone()
            print("Item ID:     "+str(R[0]))
            print("Item Name:   "+str(R[1]))
            print("Category:    "+str(R[2]))
            print("Base Price:  "+str(R[3]))
            print("Start Bidding")
            print("Opening Bid: "+str(R[3]))
            choice="Y"
            latest_bid = R[3]
            B=0
            while choice=='Y' or choice=='y':
                try:
                    bid_username=input("Enter the username of the bidder")
                    bid_user_password=input("Enter the password of the bidder")
                    A = "SELECT * FROM BIDDERS WHERE USERNAME = '"+bid_username+"'"
                    MyCur.execute(A)
                    P = MyCur.fetchone()
                    if P[4]==bid_user_password:
                        print("The  password  entered  is  correct.  Welcome,  Mr./Ms.
"+P[1]+"!")
                        bid_amt=int(input("Enter your bidding amount"))
                        if bid_amt<=latest_bid:
                            print("The bid amount entered is lesser than or equal to
the latest bid! Please try again.")
                        else:
                            print("Latest     bid     by     "+bid_username+"     for
"+str(bid_amt))
                            latest_bidder = bid_username
                            latest_bid = bid_amt
                            B+=1
                    else:
                        print("The   password   entered   is   incorrect.   Please   try
again!")
                except Exception as e:
                   print(str(e))
                   print("The username entered does not exists!")
                choice = input("Do you want to continue bidding?(Y/N)")
            if B>0:
                print("Item    "+R[1]+"    sold    for    "+str(latest_bid)+"    to
"+latest_bidder)
                A = "UPDATE  ITEMS  SET  STATUS  =  'Close',  DOP  =  CURDATE(),  BUYER  =
'"+P[1]+"' WHERE ID="+I
                MyCur.execute(A)
                A       =       "UPDATE       BIDDERS       SET       QTY=QTY+1,
TOTAL_AMOUNT=TOTAL_AMOUNT+"+str(latest_bid)+", DOP=CURDATE()"
```

```
                MyCur.execute(A)
                A = "INSERT INTO " + latest_bidder + " VALUES(" + str(R[0]) + ", '"
+ R[1] + "', " + str(R[3]) + ", " + str(latest_bid) + ", '" + R[2] + "', CURDATE(),
'NO')"
                MyCur.execute(A)
                db.commit()
            elif B==0:
                print("Item "+R[1]+" not sold!")
        else:
            print("Item not available for bidding!")
    else:
        print("Please try again!")


'''''''''''''''''''''''''''''''''''''''''''
To generate a bill for final payment
'''''''''''''''''''''''''''''''''''''''''''

def Generate_Bill():
    username = input("Enter the username for which the payment has to be made")
    attempts = 0
    var = 'True'
    while var=='True':
        password = input("Enter the password associated to the entered username")
        S = "SELECT * FROM BIDDERS WHERE USERNAME = '"+username+"'"
        MyCur.execute(S)
        P = MyCur.fetchone()
        if P[4]==password:
            print("The password entered is correct. Welcome, Mr/Ms ",P[1],"!")
            print("The quantity of items you purchased is: ",P[5])
            print("The total amount that you need to pay is Rs.",P[6])
            A = "UPDATE BIDDERS SET QTY = 0, TOTAL_AMOUNT = 0, DOP = CURDATE()"
            MyCur.execute(A)
            A = "UPDATE "+username+" SET BILL_PAID = 'Yes' WHERE BILL_PAID = 'No'"
            MyCur.execute(A)
            db.commit()
            print("The  payment  has  been  done.  Thank  you  Mr./Ms.  ",P[1],"  for
shopping with us!")
            var='False'
        else:
            print("The password entered is wrong!")
            attempts+=1
        if attempts==3:
            print("The number of wrong password attempts has been exhausted. Please
try again after some time!")
            var = 'False'
```

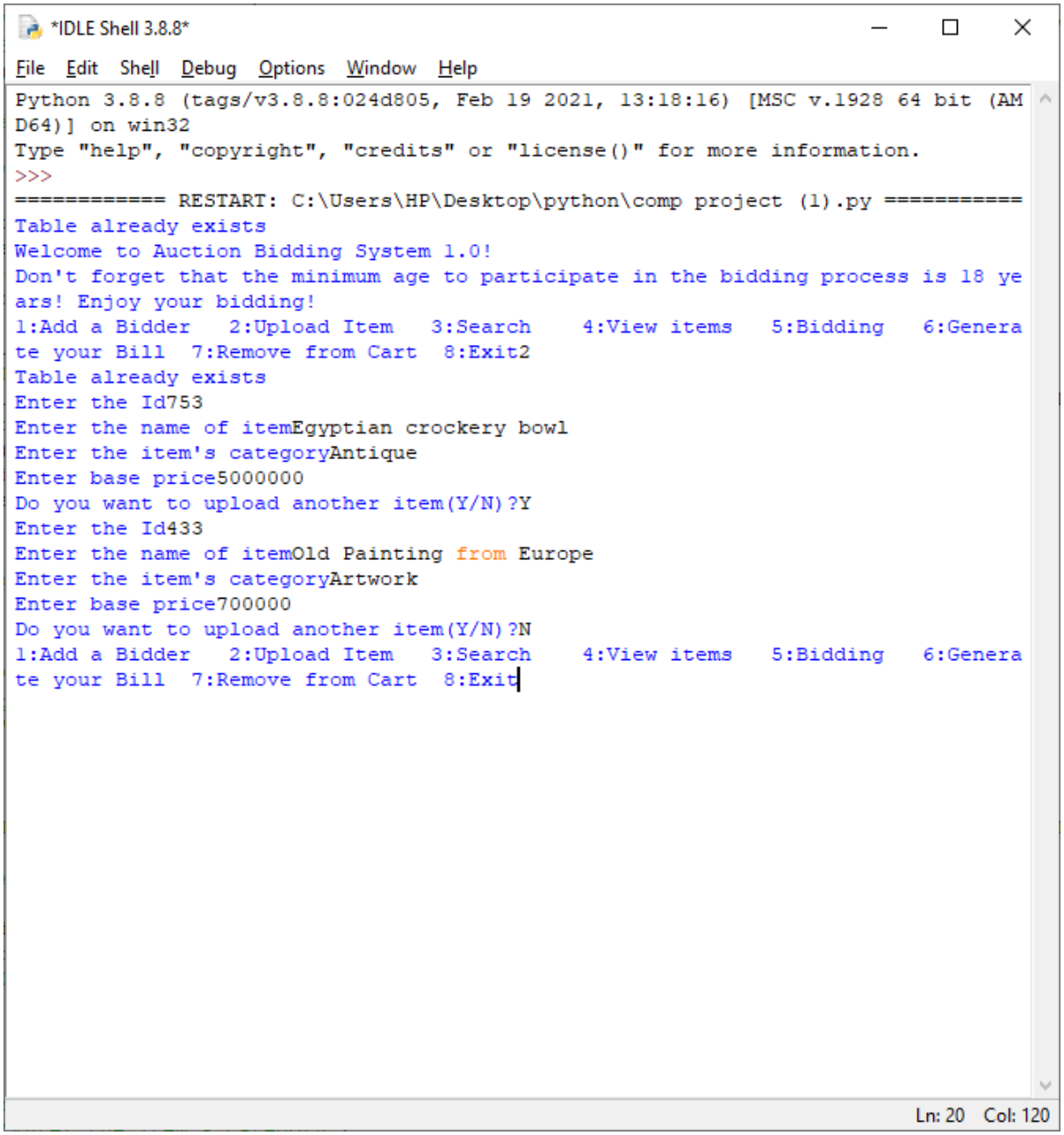
```
''''''''''''''''''''''''''''''''''''''''''''
To remove any items from the cart
''''''''''''''''''''''''''''''''''''''''''''

def Remove_from_Cart():
    username = input("Enter the username for which the cart has to be modified.")
    attempts = 0
    var = 'True'
    while var == 'True':
        password = input("Enter the password associated to the entered username")
        S = "SELECT * FROM BIDDERS WHERE USERNAME = '"+username+"'"
        MyCur.execute(S)
        P = MyCur.fetchone()
        if P[4]==password:
            print("The password entered is correct. Welcome, Mr/Ms "+P[1]+"!")
            SQL = "SELECT * FROM "+username+" WHERE BILL_PAID  = 'NO'"
            MyCur.execute(SQL)
            R = MyCur.fetchall()
            for i in R:
                print(i)
            item_id = int(input("Please enter the ID of the item that you need to
remove."))
            SQL = "SELECT * FROM "+username+" WHERE ID =",item_id
            MyCur.execute(SQL)
            A = MyCur.fetchone()
            purchase_price = A[3]
            SQL = "DELETE FROM "+username+" WHERE ID =",item_id
            MyCur.execute(SQL)
            SQL = "UPDATE ITEMS SET BID_AMOUNT = '', STATUS = 'Open', DOP = '',
Buyer = '' WHERE ID = ",item_id
            MyCur.execute(SQL)
            SQL = "UPDATE BIDDERS SET QTY=QTY-1, TOTAL_AMOUNT = TOTAL_AMOUNT -
",purchase_price," WHERE USERNAME = '"+username+"'"
            db.commit()
            S = "SELECT * FROM BIDDERS WHERE USERNAME = '"+username+"'"
            MyCur.execute(S)
            R = MyCur.fetchone()
            print("The quantity of items in your cart now are: ",R[5])
            print("The total amount that you now need to pay is Rs.",[6])
            var = 'False'
    else:
        print("The password entered is wrong!")
        attempts+=1
        if attempts==3:
            print("The number of wrong password attempts has been exhausted. Please
try again after some time!")
            var = 'False'
```

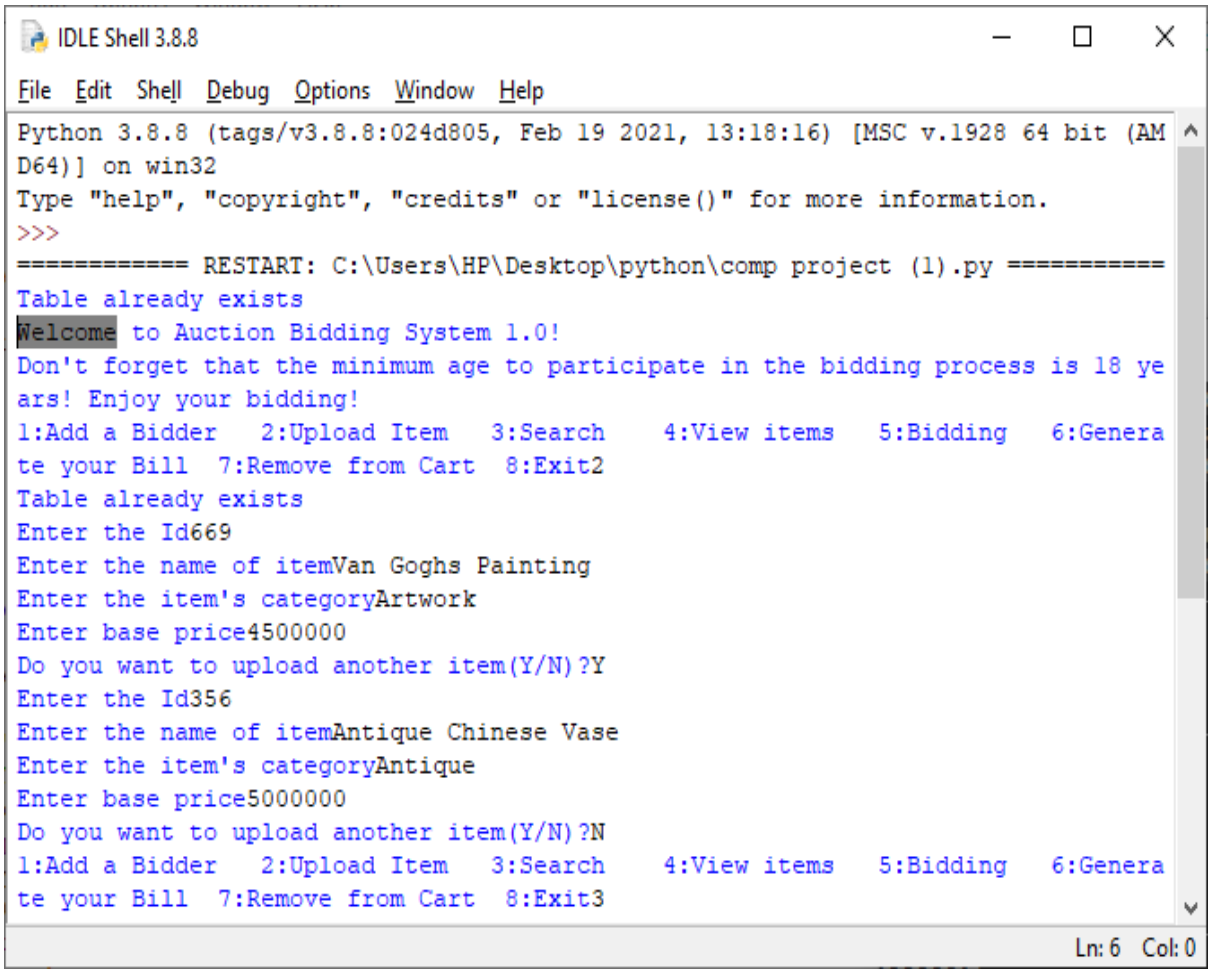
```
''''''''''''''''''''''''''''''''''''''''''''
Main Program (User-driven)
''''''''''''''''''''''''''''''''''''''''''''

print("Welcome to Auction Bidding System 1.0!")
print("Don't forget that the minimum age to participate in the bidding process is 18
years! Enjoy your bidding!")
while True:
   Option = input("1:Add a Bidder   2:Upload Item   3:Search    4:View items/bidder
5:Bidding   6:Generate your Bill  7:Remove from Cart  8:Exit")
   if Option=="1":
      Add_Bidder()
   elif Option=="2":
      Upload_Item()
   elif Option=="3":
      SearchItem()
   elif Option=="4":
    a=input("Do you want to view bidder or item(B/I)")
    if a=="B":
       ViewBidder()
    elif a=="I":
      opt   =   input("Do   you   want   to   view   items   open   or   closed   for
bidding?(Open/Closed)")
      View(opt)
   elif Option=="5":
      Bidding()
   elif Option=="6":
      Generate_Bill()
   elif Option=="7":
      Remove_from_Cart()
   elif Option=="8":
      print("Thank you for visiting!")
      break
   else:
      print("Please enter a valid option!")
db.close()
```

ADD BIDDER

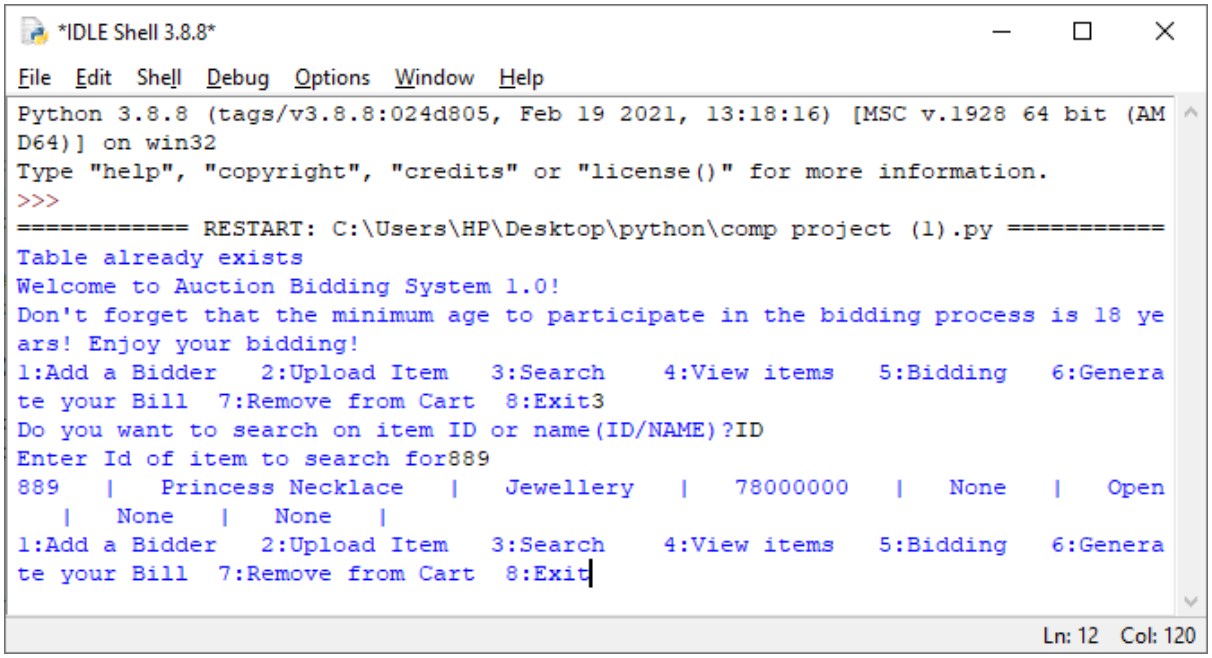

IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\python\comp project (1).py ==========
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit1
Enter name of bidderGirish Sharma
Enter username of bidder (The username can only be a combination of alphabets, numbers and underscores without any spaces)girishsharma34
Enter age of bidder66
Enter passwordgirish@234
Enter you email idgirishsharma34@gmail.com
Enter your mobile number8898955666
Table created
Bidder added
Do you want to add another bidder(Y/N)?Y
Enter name of bidderRajeev Prakash
Enter username of bidder (The username can only be a combination of alphabets, numbers and underscores without any spaces)rajeevp123
Enter age of bidder54
Enter passwordrajeevp@123
Enter you email idrajeevprakash@gmail.com
Enter your mobile number9910804476
Table created
Bidder added
Do you want to add another bidder(Y/N)?N
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit
Ln: 27 Col: 120

UPLOAD ITEM

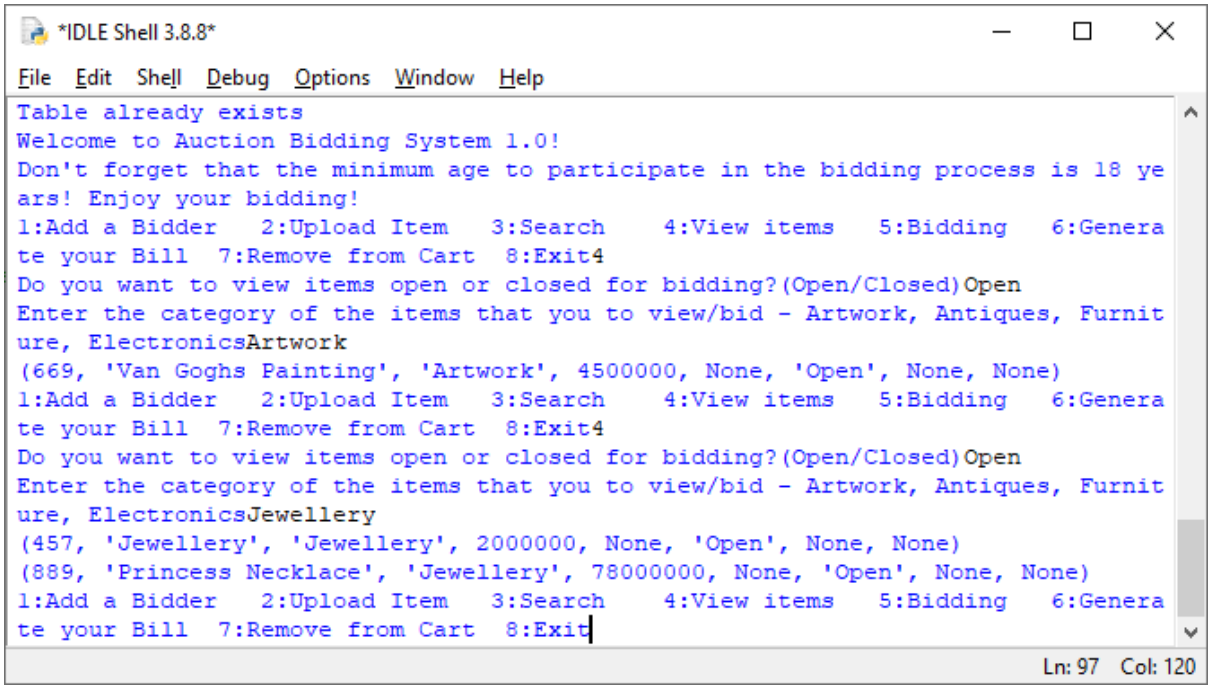

IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\python\comp project (1).py ==========
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit2
Table already exists
Enter the Id753
Enter the name of itemEgyptian crockery bowl
Enter the item's categoryAntique
Enter base price5000000
Do you want to upload another item(Y/N)?Y
Enter the Id433
Enter the name of itemOld Painting from Europe
Enter the item's categoryArtwork
Enter base price700000
Do you want to upload another item(Y/N)?N
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit
Ln: 20 Col: 120

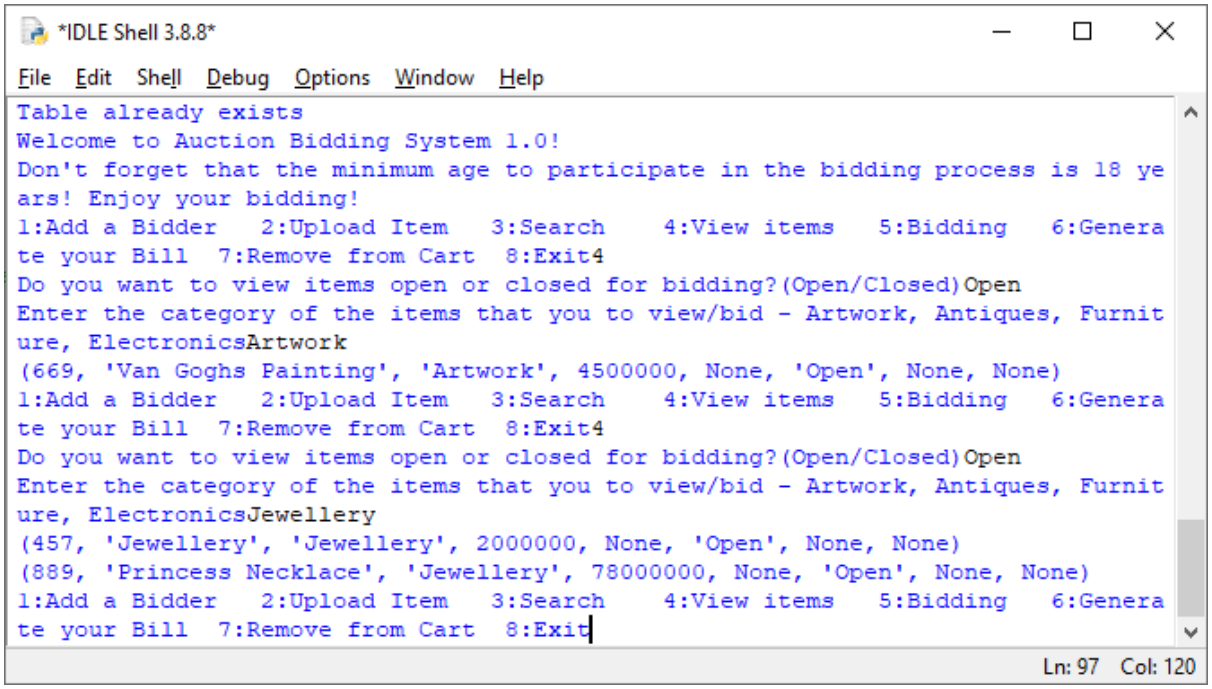

IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\python\comp project (1).py ==========
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit2
Table already exists
Enter the Id669
Enter the name of itemVan Goghs Painting
Enter the item's categoryArtwork
Enter base price4500000
Do you want to upload another item(Y/N)?Y
Enter the Id356
Enter the name of itemAntique Chinese Vase
Enter the item's categoryAntique
Enter base price5000000
Do you want to upload another item(Y/N)?N
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit3
Ln: 6 Col: 0

SEARCH ITEM

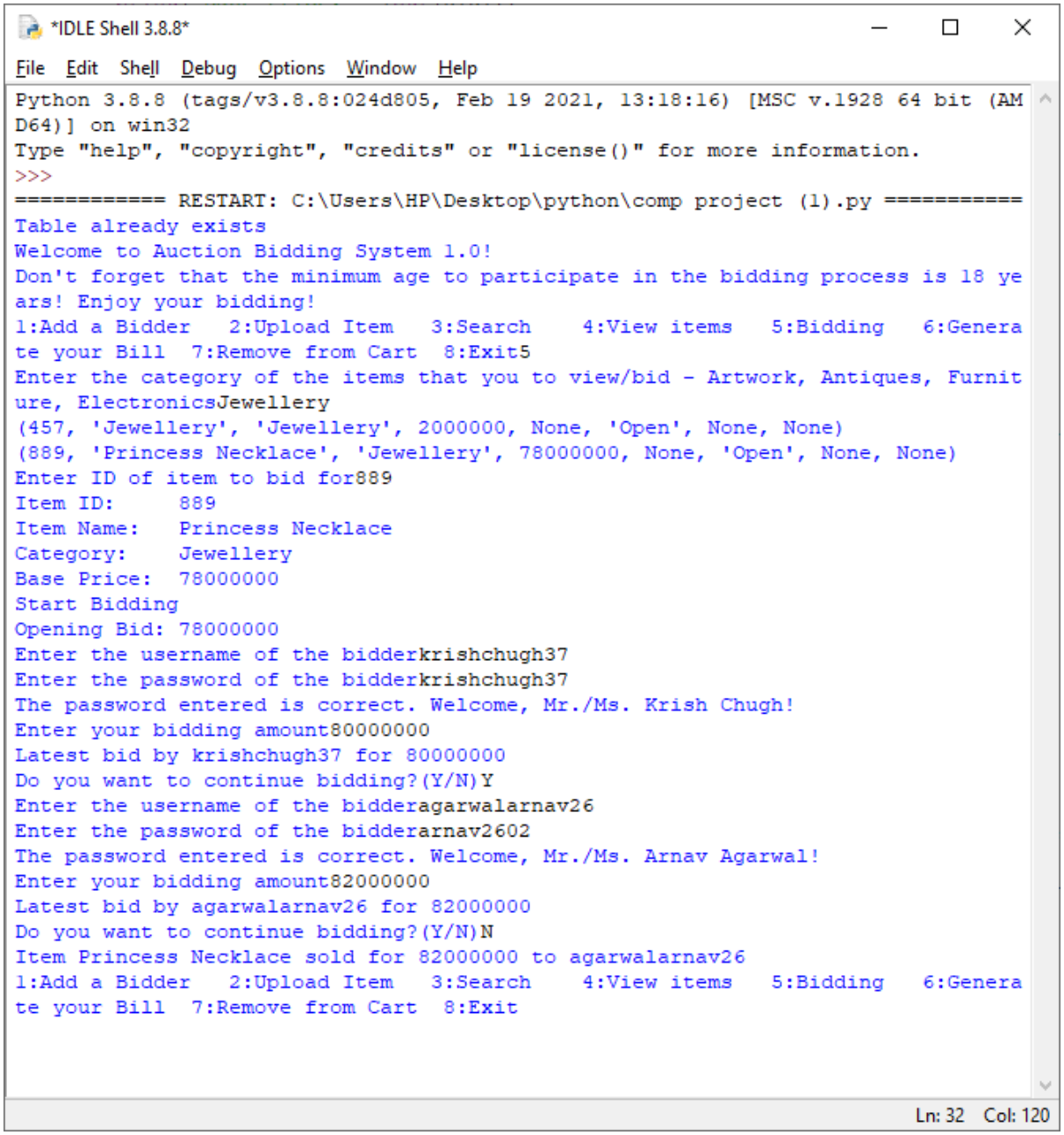

IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\python\comp project (1).py ==========
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit3
Do you want to search on item ID or name(ID/NAME)?ID
Enter Id of item to search for889
889 | Princess Necklace | Jewellery | 78000000 | None | Open | None | None |
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit
Ln: 12 Col: 120

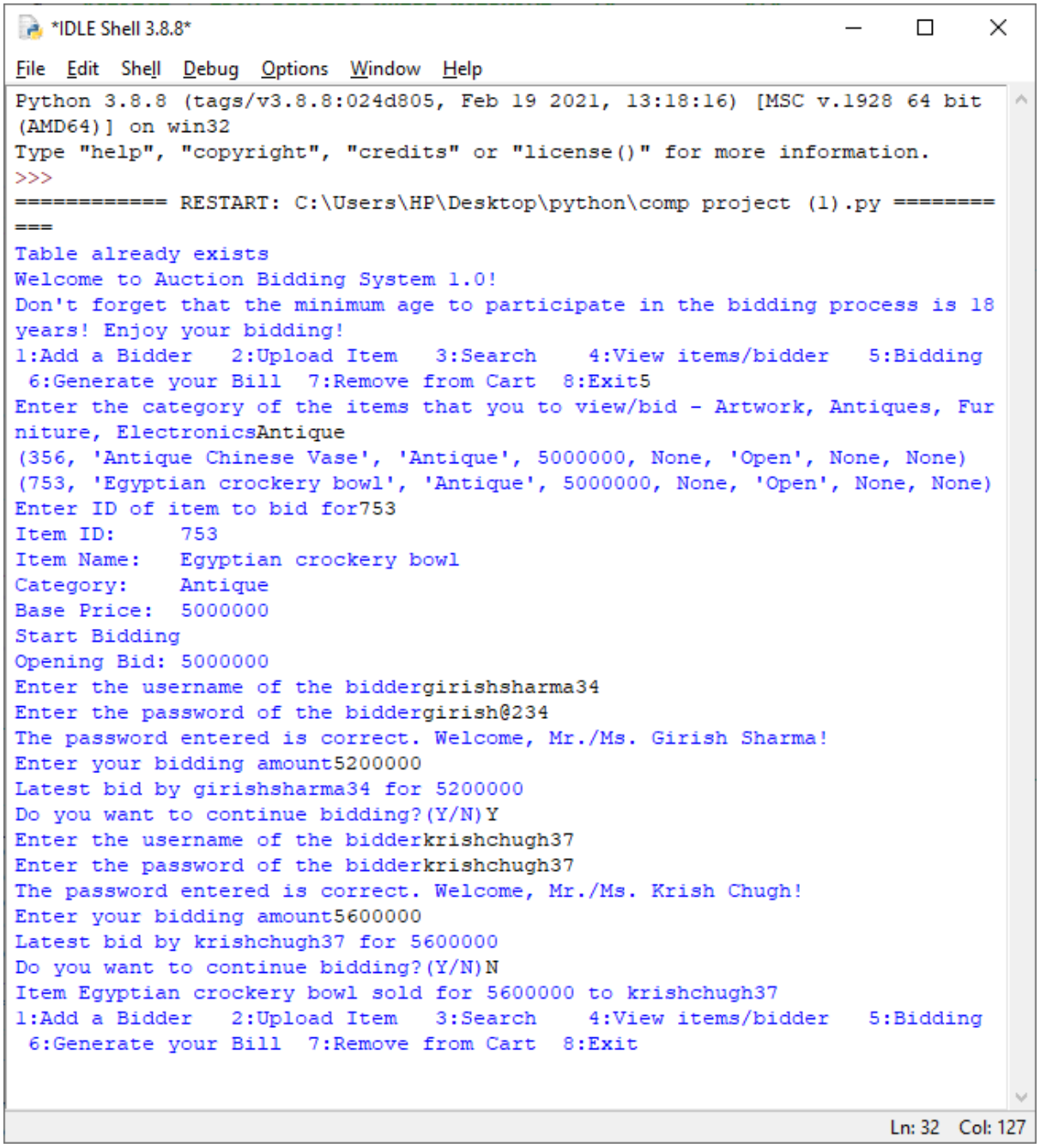
VIEW ITEM/BIDDER



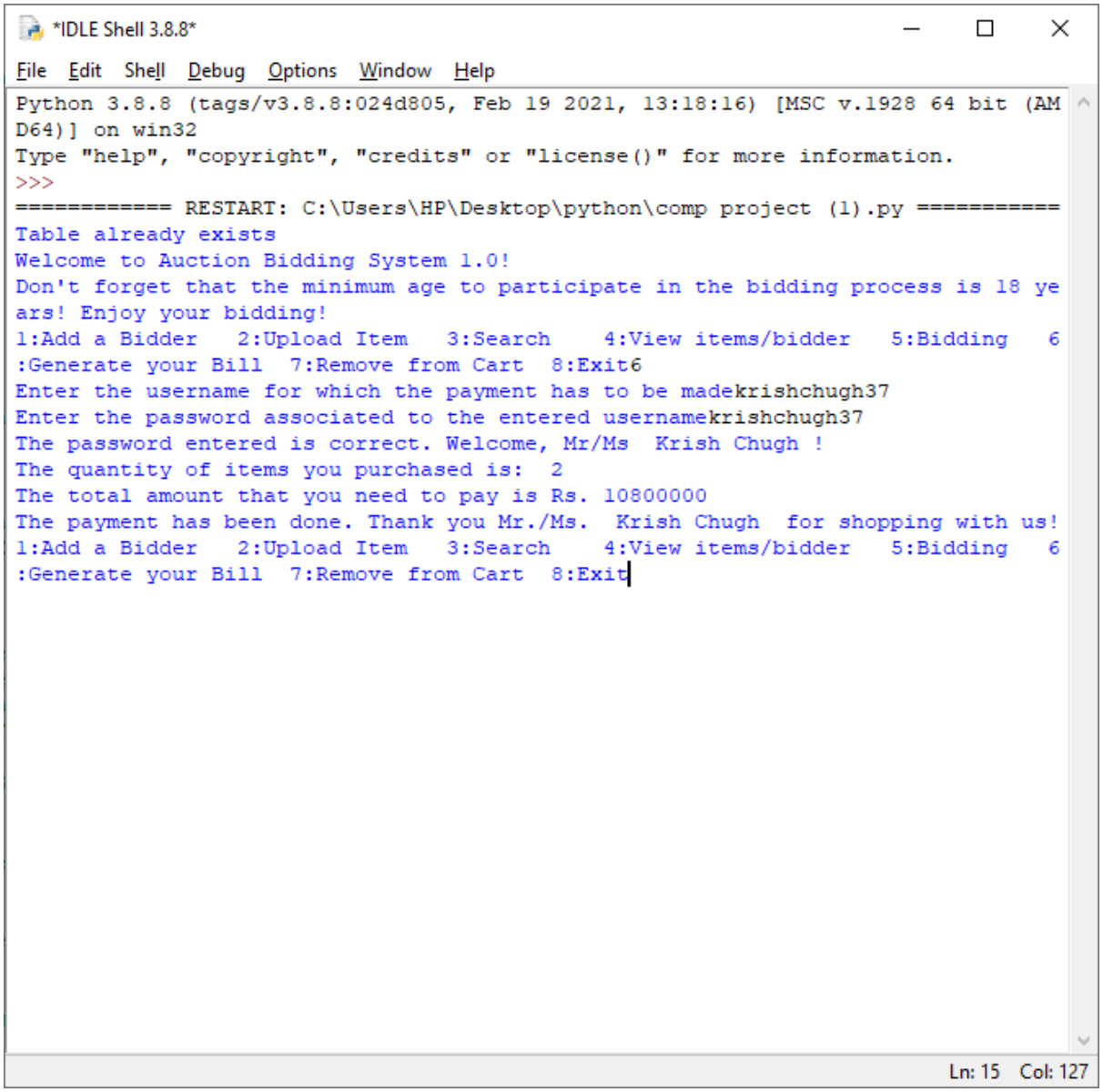


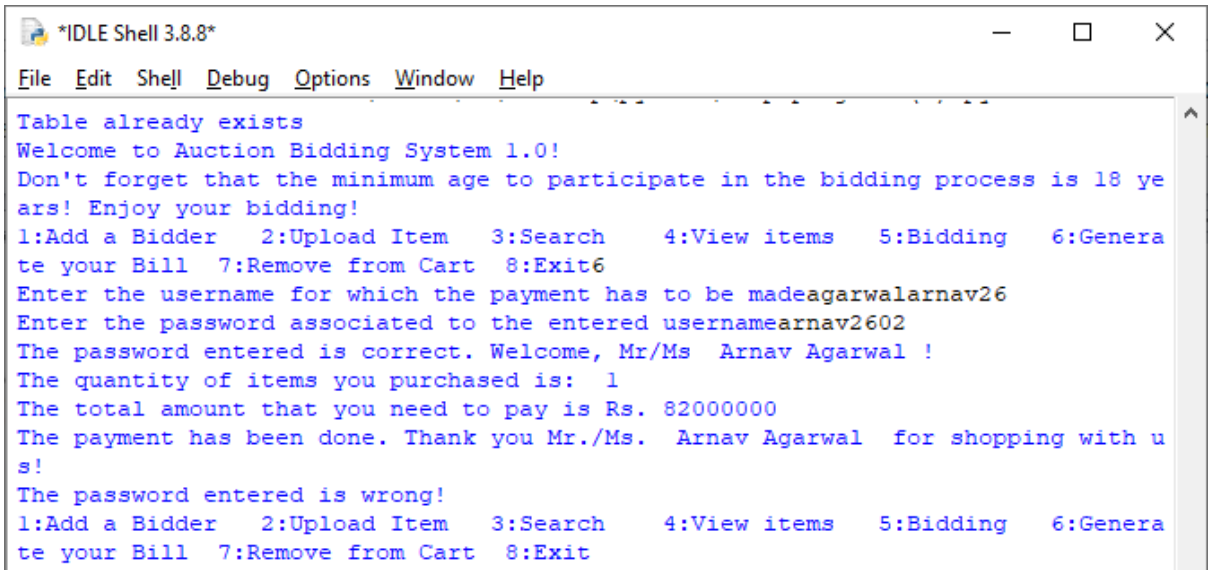
BIDDING


IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\python\comp project (1).py ==========
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit5
Enter the category of the items that you to view/bid - Artwork, Antiques, Furniture, ElectronicsJewellery
(457, 'Jewellery', 'Jewellery', 2000000, None, 'Open', None, None)
(889, 'Princess Necklace', 'Jewellery', 78000000, None, 'Open', None, None)
Enter ID of item to bid for889
Item ID: 889
Item Name: Princess Necklace
Category: Jewellery
Base Price: 78000000
Start Bidding
Opening Bid: 78000000
Enter the username of the bidderkrishchugh37
Enter the password of the bidderkrishchugh37
The password entered is correct. Welcome, Mr./Ms. Krish Chugh!
Enter your bidding amount80000000
Latest bid by krishchugh37 for 80000000
Do you want to continue bidding?(Y/N)Y
Enter the username of the bidderagarwalarnav26
Enter the password of the bidderarnav2602
The password entered is correct. Welcome, Mr./Ms. Arnav Agarwal!
Enter your bidding amount82000000
Latest bid by agarwalarnav26 for 82000000
Do you want to continue bidding?(Y/N)N
Item Princess Necklace sold for 82000000 to agarwalarnav26
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit
Ln: 32 Col: 120


IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\python\comp project (1).py ==========
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items/bidder 5:Bidding
6:Generate your Bill 7:Remove from Cart 8:Exit5
Enter the category of the items that you to view/bid - Artwork, Antiques, Furniture, ElectronicsAntique
(356, 'Antique Chinese Vase', 'Antique', 5000000, None, 'Open', None, None)
(753, 'Egyptian crockery bowl', 'Antique', 5000000, None, 'Open', None, None)
Enter ID of item to bid for753
Item ID: 753
Item Name: Egyptian crockery bowl
Category: Antique
Base Price: 5000000
Start Bidding
Opening Bid: 5000000
Enter the username of the biddergirishsharma34
Enter the password of the biddergirish@234
The password entered is correct. Welcome, Mr./Ms. Girish Sharma!
Enter your bidding amount5200000
Latest bid by girishsharma34 for 5200000
Do you want to continue bidding?(Y/N)Y
Enter the username of the bidderkrishchugh37
Enter the password of the bidderkrishchugh37
The password entered is correct. Welcome, Mr./Ms. Krish Chugh!
Enter your bidding amount5600000
Latest bid by krishchugh37 for 5600000
Do you want to continue bidding?(Y/N)N
Item Egyptian crockery bowl sold for 5600000 to krishchugh37
1:Add a Bidder 2:Upload Item 3:Search 4:View items/bidder 5:Bidding
6:Generate your Bill 7:Remove from Cart 8:Exit
Ln: 32 Col: 127

GENERATE BILL




IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Table already exists
Welcome to Auction Bidding System 1.0!
Don't forget that the minimum age to participate in the bidding process is 18 years! Enjoy your bidding!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit6
Enter the username for which the payment has to be madeagarwalarnav26
Enter the password associated to the entered usernamearnav2602
The password entered is correct. Welcome, Mr/Ms Arnav Agarwal !
The quantity of items you purchased is: 1
The total amount that you need to pay is Rs. 82000000
The payment has been done. Thank you Mr./Ms. Arnav Agarwal for shopping with us!
The password entered is wrong!
1:Add a Bidder 2:Upload Item 3:Search 4:View items 5:Bidding 6:Generate your Bill 7:Remove from Cart 8:Exit

Hardware & Software Requirement

Hardware Requirement

PC/Laptop/MacBook with
Intel core/i3/i5/i7 or any equivalent
With at least 2 GB RAM
10 MB free space on Hard Disk
LCD/LED

Operating System & Compiler

MS Windows/MacOS

Python IDLE 3.x
OR
colab.research.google.com (gmail account)

and/or
MySQL 8.x