PIZZA RETAIL SYSTEM

PROJECT J.J.J.

- **ARYAN JAIN**
  MANAV JAIN
  SIDDHARATHA JAISWAL

CONTENTS

SYNOPSIS

Nowadays, pizza delivery systems of various global food chains is becoming increasingly incompetent and redundant. Thus, we decided to create a pizza delivery system which will be retail-based and will allow the company to place orders, modify menu and keep a tab on inventory.

INPUTS

FUNCTION 1:
Main Menu

- Place Order
- Modify Menu and View Inventory

FUNCTION 2:
Place an Order

- Name
- Address
- Quantity of Same Pizza
- Size
- Crust
- Cheese
- Topping
- Order more different pizzas

FUNCTION 3:

Modify Menu and View Inventory

- View Inventory
- Add or replace new product
- Update cost price or selling price

OUTPUTS

FUNCTION 1:
Main Menu

- Function entered

FUNCTION 2:
Place an Order

- Price of the final order.

FUNCTION 3:
Modify Menu and View Inventory

- Inventory Table

CONCEPTS

Concepts Learned In Class:

- Basics of python
- Basics of SQL

Concepts Learned On Our Own:

- Python-SQL integration

INTIALISE PROGRAM

```
import sqlite3

connection = sqlite3.connect("Order_Placed.db")

crsr = connection.cursor()


# Creating Inventory and Menu Table
sql_command = """CREATE TABLE inventory (
menu_id INTEGER NOT NULL,
ingredient VARCHAR(20) NOT NULL,
qty INTEGER(3) NOT NULL,
cost_price integer(4) NOT NULL,
```

```
sell_price integer(4) not null);"""

crsr.execute(sql_command)

# Inserting default values into Inventory table
sql_command = """INSERT INTO inventory
VALUES
(1, '10" - Thin', 100, 60 , 69),
(2, '10" - Regular', 100, 50 , 57.5),
(3, '10" - Cheezy', 100, 70 , 80.5),
(4, '12" - Thin', 100, 70 , 69),
(5, '12" - Regular', 100, 60 , 57.5),
(6, '12" - Cheezy', 100, 80 , 80.5),

(7, '14" - Thin', 100, 80 , 69),
(8, '14" - Regular', 100, 70 , 57.5),
(9, '14" - Cheezy', 100, 90 , 80.5),

(10, "Cheese -> Fat - Free" , 100, 50, 57.5),
(11, "Cheese -> Vegan", 100, 40, 46),
```

```
(12, "Cheese -> Extra", 100, 30, 34.5),
(13, "Cheese -> Regular", 100, 20, 23),

(14, "Onion", 100, 10 , 11.5 ),
(15, "Olives", 100, 10 , 11.5 ),
(16, "Jalapeno", 100, 15 , 17.25 ),
(17, "Red Paprika", 100, 15 , 17.25 ),
(18, "Tomato", 100, 5 , 5.75 ),
(19, "Corn", 100, 5 , 5.75 ),
(20, "Capsicum", 100, 5 , 5.75 ),
(21, "Mushroom", 100, 10 , 11.5 );"""

crsr.execute(sql_command)

# Creating Pizza Order Table
sql_command = """CREATE TABLE pizza (
customer_id INTEGER PRIMARY KEY AUTOINCREMENT,
name VARCHAR(20) NOT NULL,
address VARCHAR(100) NOT NULL,
pizza_size INTEGER(2) NOT NULL,
```

```
crust CHAR(7) NOT NULL,
topping_1 CHAR(11),
topping_2 CHAR(11),
topping_3 CHAR(11),
cheese CHAR(10) NOT NULL,
quantity INTEGER(2) NOT NULL,
order_date DATE NOT NULL);"""

crsr.execute(sql_command)


connection.commit()

connection.close()
```

MAIN MENU PROGRAM

```
import sqlite3

from datetime import datetime

from datetime import date

from datetime import timedelta



def prRed(skk): print("\033[91m
{}\033[00m" .format(skk))

def prGreen(skk): print("\033[92m
{}\033[00m" .format(skk))

def prYellow(skk): print("\033[93m
{}\033[00m" .format(skk))

def prLightPurple(skk): print("\033[94m
{}\033[00m" .format(skk))
```

```
def prPurple(skk): print("\033[95m
{}\033[00m" .format(skk))

def prCyan(skk): print("\033[96m
{}\033[00m" .format(skk))

def prLightGray(skk): print("\033[97m
{}\033[00m" .format(skk))

def prBlack(skk): print("\033[98m
{}\033[00m" .format(skk))


def main_menu():


        while True:

            string = "***** WELCOME TO J.J.J. PIZZA SYSTEM
(ALPHA VERSION 2.0) *****".center(160, ' ')


            print('\n')

            prYellow(string)


            string1 = '''           Brought to you by -
```

```
| ARYAN JAIN | SIDDHARATHA JAISWAL | MANAV JAIN
|'''.center(300, ' ')


        print('\n')

        prLightGray(string1)


        prCyan("""\n\nSelect a command below \n\n1. Place
an order \n2. View inventory and modify menu  """)

        prLightPurple('\nEnter the function based on the
numbers above:  ')

        print()

        Main_choice = input()



        if Main_choice=="1":

            connection = sqlite3.connect("Order_Placed.db")

            crsr = connection.cursor()

            def order():

                    Sum=0
```

```
                string1 = "***** PLACE AN ORDER
*****".center(160, ' ')
                print('\n')
                prYellow(string1)
            #NAME
                while True:
                    prCyan('\nEnter Name: ')
                    print()
                    Name = input()
                    if Name.isdigit() or Name == "":
                        prRed("\nInvalid Name")
                    else:
                        break
            #ADDRESS
                while True:
                    prCyan('\nEnter Address: ')
                    print()
                    Address = input()
                    if Address != "":
                        break
```

```
                else:
                    prRed('\nInvalid Address')
            #QUANTITY
                while True:
                try:
                    prCyan('\n\n\nHow many pizzas of
the same type would you be ordering: ')
                    print()
                    Qty = float(input())
                except:
                    prRed("\nInvalid Quantity")
                else:
                    break


            #SIZE AND CRUST
                prCyan('''\n\n\n\nChoose a size below\n
\n1. 10\" \n2. 12\" \n3. 14\"''')
                while True:
                    prPurple('\nEnter size based on the
numbers above: ')
                    print( )
```

```
                    Size = input()
                    if Size == '1':
                        Size = 10
                        prCyan('''\n\n\n\nChoose a crust
below \n\n1. Thin \n2. Regular \n3. Cheezy''')
                        while True:
                            prPurple('\nEnter crust based
on the numbers above: ')
                            print()
                            Crust = input()
                            if Crust == "1":
                                Crust = "Thin"
                                crsr =
connection.execute("SELECT sell_price from inventory
WHERE menu_id=1")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=1''')
                                break
                            elif Crust == "2":
                                Crust = "Regular"
```

```
                                        crsr =
connection.execute("SELECT Sell_price from inventory
WHERE menu_id=2")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=2''')
                                        break
                                    elif Crust == "3":
                                        Crust = "Cheezy"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=3")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=3''')
                                        break
                                    elif Crust == "":
                                        prRed("\nInvalid Crust")
                                    else:
                                        prRed('\nInvalid Crust')
                                rows = crsr.fetchall()
                                for row in rows:
```

```
                    Temp = int(row[0])
                    Sum += Temp
                break
            elif Size == '2':
                Size = 12
                prCyan('''\n\n\n\nChoose a crust
below \n\n1. Thin \n2. Regular \n3. Cheezy''')
                while True:
                    prPurple('\nEnter crust based
on the numbers above: ')
                    print()
                    Crust = input()
                    if Crust == "1":
                        Crust = "Thin"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=4")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=4''')
                        break
                    elif Crust == "2":
```

```
                                    Crust = "Regular"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=5")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=5''')
                                        break
                                    elif Crust == "3":
                                        Crust = "Cheezy"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=6")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=6''')
                                        break
                                    elif Crust == "":
                                        prRed("\nInvalid Crust")
                                    else:
                                        prRed('\nInvalid Crust')
                                rows = crsr.fetchall()
```

```
                    for row in rows:
                        Temp = int(row[0])
                        Sum += Temp
                    break
                elif Size == '3':
                    Size = 14
                    prCyan('''\n\n\n\nChoose a crust
below \n\n1. Thin \n2. Regular \n3. Cheezy''')
                    while True:
                        prPurple('\nEnter crust based
on the numbers above: ')
                        print()
                        Crust = input()
                        if Crust == "1":
                            Crust = "Thin"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=7")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=7''')
                        break
```

```
                elif Crust == "2":
                    Crust = "Regular"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=8")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=8''')
                    break
                elif Crust == "3":
                    Crust = "Cheezy"

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=9")

connection.execute(f'''UPDATE inventory SET qty = qty-
{Qty}  WHERE menu_id=9''')
                    break
                elif Crust == "":
                    prRed("\nInvalid Crust")
                else:
                    prRed('\nInvalid Crust')
```

```
                rows = crsr.fetchall()
                for row in rows:
                    Temp = int(row[0])
                    Sum += Temp
                break
            elif Size == "":
                prRed('\nInvalid Size')
            else:
                prRed('\nInvalid size')

        #CHEESE
            prCyan('''\n\n\n\nChoose a cheese below \n\n1. Fat-free \n2. Vegan \n3. Extra \n4. Regular''')
            while True:
                prPurple('\nEnter cheese based on the numbers above: ')
                print()
                Cheese = input()
                if Cheese == '1':
                    Cheese = "Fat-free"
```

```
                    crsr=connection.execute("SELECT
Sell_price from inventory WHERE menu_id=10")

                    connection.execute(f'''UPDATE
inventory SET qty = qty-{Qty}  WHERE menu_id=10''')

                    break

                elif Cheese == '2':

                    Cheese = "Vegan"

                    crsr=connection.execute("SELECT
Sell_price from inventory WHERE menu_id=11")

                    connection.execute(f'''UPDATE
inventory SET qty = qty-{Qty}  WHERE menu_id=11''')

                    break

                elif Cheese == '3':

                    Cheese = "Extra"

                    crsr=connection.execute("SELECT
Sell_price from inventory WHERE menu_id=12")

                    connection.execute(f'''UPDATE
inventory SET qty = qty-{Qty}  WHERE menu_id=12''')

                    break

                elif Cheese == '4':

                    Cheese = "Regular"
```

```
                    crsr=connection.execute("SELECT
Sell_price from inventory WHERE menu_id=13")
                    connection.execute(f'''UPDATE
inventory SET qty = qty-{Qty}  WHERE menu_id=13''')
                    break
                elif Cheese == "":
                    prRed('\nInvalid Cheese')
                else:
                    prRed('\nInvalid Cheese')
            rows = crsr.fetchall()
            for row in rows:
                Temp = int(row[0])
                Sum += Temp
            print('\n\n\n')


        #TOPPINGS
            Toppings = {1:"Onion", 2:"Olives" ,
3:"Jalapeno", 4:"Red Paprika" , 5:"Tomato" , 6:"Corn" ,
7:"Capsicum" , 8:"Mushroom" ,
                    9:None}
```

```
                    cursor = connection.execute("SELECT
ingredient from inventory where menu_id between 14 and
21")

                    Tempo = 1

                    prCyan("Choose toppings from below: (Upto
3)")

                    for row in cursor:

                        prCyan(str(Tempo) + ". " + row[0])

                        Tempo += 1

                    while True:

                            Topping = []

                            i = 0

                            while i<3:

                                prPurple('\nEnter toppings based
on the numbers above: ')

                                print()

                                Topping_input = input()

                                if Topping_input == "1" or
Topping_input == "2" or Topping_input == "3" or
Topping_input == "4" or Topping_input == "5" or
Topping_input == "6" or Topping_input == "7" or
Topping_input == "8" or Topping_input == "9":
```

```
                                Topping_input =
int(Topping_input)
                                Topping_input += 13
                                crsr.execute(f'''UPDATE
inventory SET qty = qty- {Qty}  WHERE
menu_id={str(Topping_input)}''')

crsr=connection.execute("SELECT Sell_price from inventory
WHERE menu_id=" + str(Topping_input))
                                rows = crsr.fetchall()
                                for row in rows:
                                    Temp=row[0]
                                    Sum+=Temp

Topping.append(Toppings[int(Topping_input)-13])
                                i+=1
                            else:
                                prRed("\nInvalid Topping")
                        break
                Sum*=Qty
                now = datetime.now()
```

```
                    current_time =
now.strftime("%H:%M+30:%S")

            #MONEY -> AMOUNT DUE

                    # entering the values into the SQL table

                    # TODO: Possible error here with SQL
connection?

                    crsr.execute("INSERT INTO pizza
(name,address,pizza_size,crust,topping_1,topping_2,topping
_3,cheese,quantity,order_date)
VALUES(?,?,?,?,?,?,?,?,?,?)",(Name,Address,Size,Crust,Toppi
ng[0],Topping[1],Topping[2],Cheese,Qty,now))

                    Sum *= Qty

                    prGreen(f'''\n\nYour total amount due is:
Rs{Sum}''')



            #MORE PIZZAS

                    while True:

                        prCyan("\n\n\nDo you want to order
more pizzas that are different \n\nEnter 'YES' or 'NO':  ")

                        print()

                        diff  = input()
```

```
                    if diff.lower() == "yes":
                        print("\n\n\n\n")
                        order()
                    elif diff.lower() == "no":
                        time_30 = datetime.now() +
timedelta(minutes=30)
                        final_time = time_30.isoformat(' ',
'seconds')
                        prGreen(f'''\n\nYour order will be
deliverd to you within 30 minutes at {final_time} \n\nHave
a good day. \nThank you.''')
                        break
                    else:
                        prRed("\nInvalid Answer")
                        continue
                    break
            order()
            sql_command = '''SELECT ingredient, sell_price
from inventory, pizza
                WHERE pizza.customer_id = (SELECT
MAX(customer_id) from pizza) and crust = ingredient or
```

```
topping_1 = ingredient or topping_2 = ingredient or
topping_3 = ingredient or cheese = ingredient
                '''
            cursor = connection.execute(sql_command)
            connection.commit()
            connection.close()
            # Enter whole code of order


        elif Main_choice=="2":


            string1 = "***** VIEW INVENTORY & MODIFY
MENU *****".center(160, ' ')
            print('\n')
            prYellow(string1)
            # id, ingredient, qty
            def inventory():
                connection =
sqlite3.connect("Order_Placed.db")
                connection.execute(f'''UPDATE inventory
SET qty= 100 WHERE qty<=15''')
```

```
                crsr=connection.execute("SELECT menu_id,
ingredient, qty from INVENTORY")
                rows = crsr.fetchall()
                for row in rows:
                    if row[2]<=15:
                        print(f'''Quantity of {row[1]} is less
than 15. Ordering more''')


                while True:
                    print('\n\n')
                    prCyan("----------------------------------
-----------------------------------")
                    prCyan("|  ID |       INGREDIENT       |
QUANTITY | COST PRICE | SELL PRICE  |")
                    prCyan("----------------------------------
-----------------------------------")
                    cursor = connection.execute("SELECT *
from inventory")
                    for row in cursor:
                        print(" |", row[0], " "*(3 -
len(str(row[0]))),
                        "|", row[1], " "*(23 - len(row[1])),
```

```
                    "|", row[2], " "*(7 - len(str(row[2]))),
                    "|", row[3], " "*(9 - len(str(row[3]))),
                    "|", row[4], " "*(9 -
len(str(row[4]))),"|")
                prCyan("----------------------------------
-----------------------------------")
                print('\n')
                prCyan("\nWould you like to modify the
menu  \n\nEnter 'YES' or 'NO':   ")
                print()
                menu_input = input()
                if menu_input.lower() == "yes" :
                    prCyan("\nPlease enter the ingredient
you would like to add: ")
                    print()
                    new_ingredient = input()
                    print('\n\n')
                    if new_ingredient.isdigit() or
new_ingredient == "":
                        prRed("\nInvalid Input")
                    prCyan("\nPlease enter the new cost
price of the ingredient: ")
```

```
                print()
                new_cp = input()
                print('\n\n')
                if new_cp.isalpha() or new_cp == "":
                    prRed("\nInvalid Input")
                prCyan("\nPlease enter the new
selling price of the ingredient: ")
                print()
                new_sp = input()
                print('\n\n')
                if new_sp.isalpha() or new_sp == "":
                    prRed("\nInvalid Input")
                prCyan("\nPlease enter the name of
the ingredient you would like to replace: ")
                print()
                old_ingredient = input()
                print('\n\n')
                if old_ingredient.isdigit() or
old_ingredient == "":
                    prRed("\nInvalid Input")
```

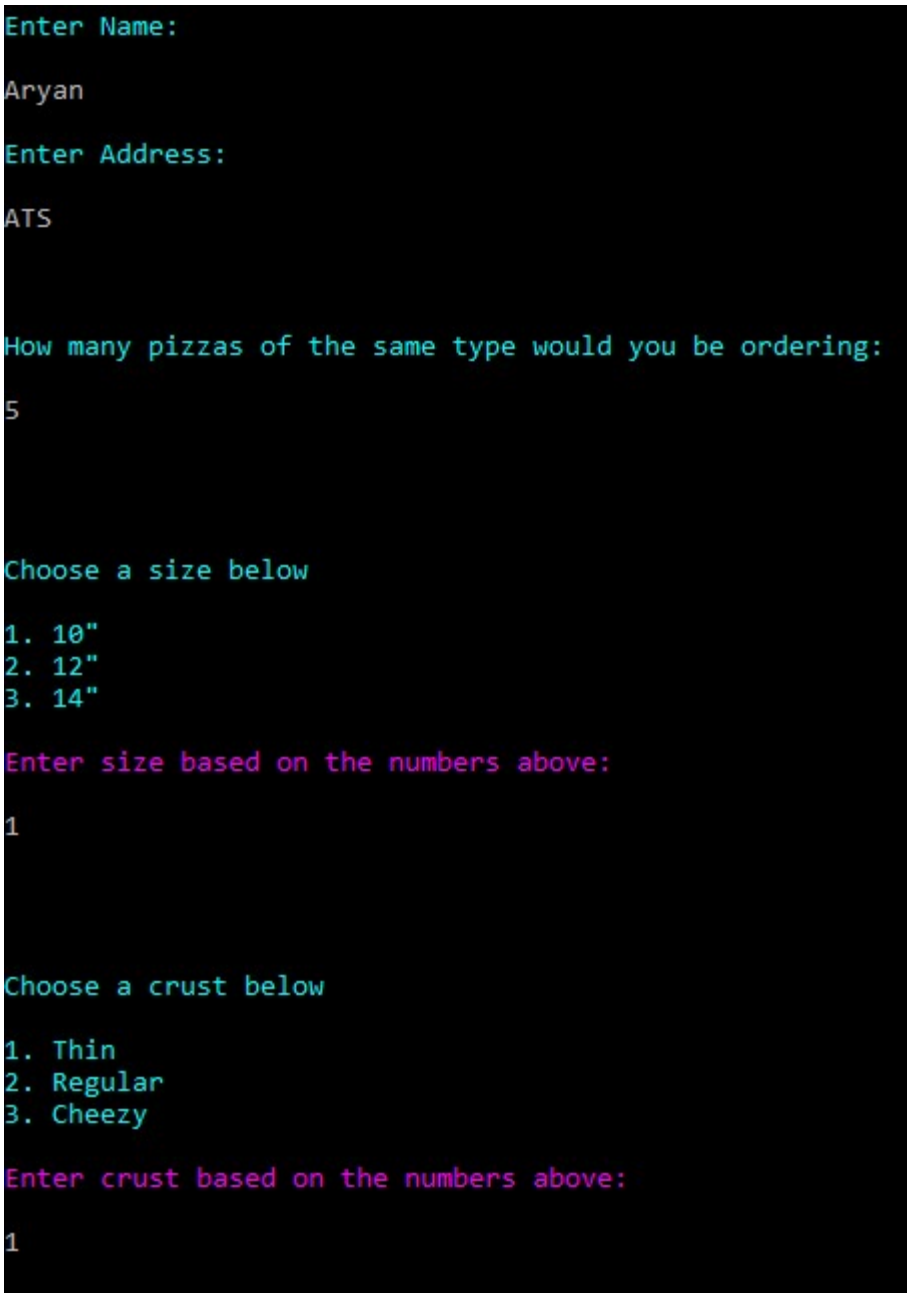
```
                        crsr.execute("UPDATE inventory SET
ingredient = ?, qty = 100, cost_price = ?, sell_price = ?
WHERE  ingredient
= ?",(new_ingredient,float(new_cp),float(new_sp),old_ingre
dient))
                    elif menu_input.lower() == "no":
                        prPurple("\n\nHave a good day.
\nThank you.")
                        break
                    else:
                        prRed("\nInvalid Input")
                connection.commit()
                connection.close()
            inventory()
        else:
            prRed("\nInvalid Input")


main_menu()
```

OUTPUT OF THE MAIN MENU PROGRAM

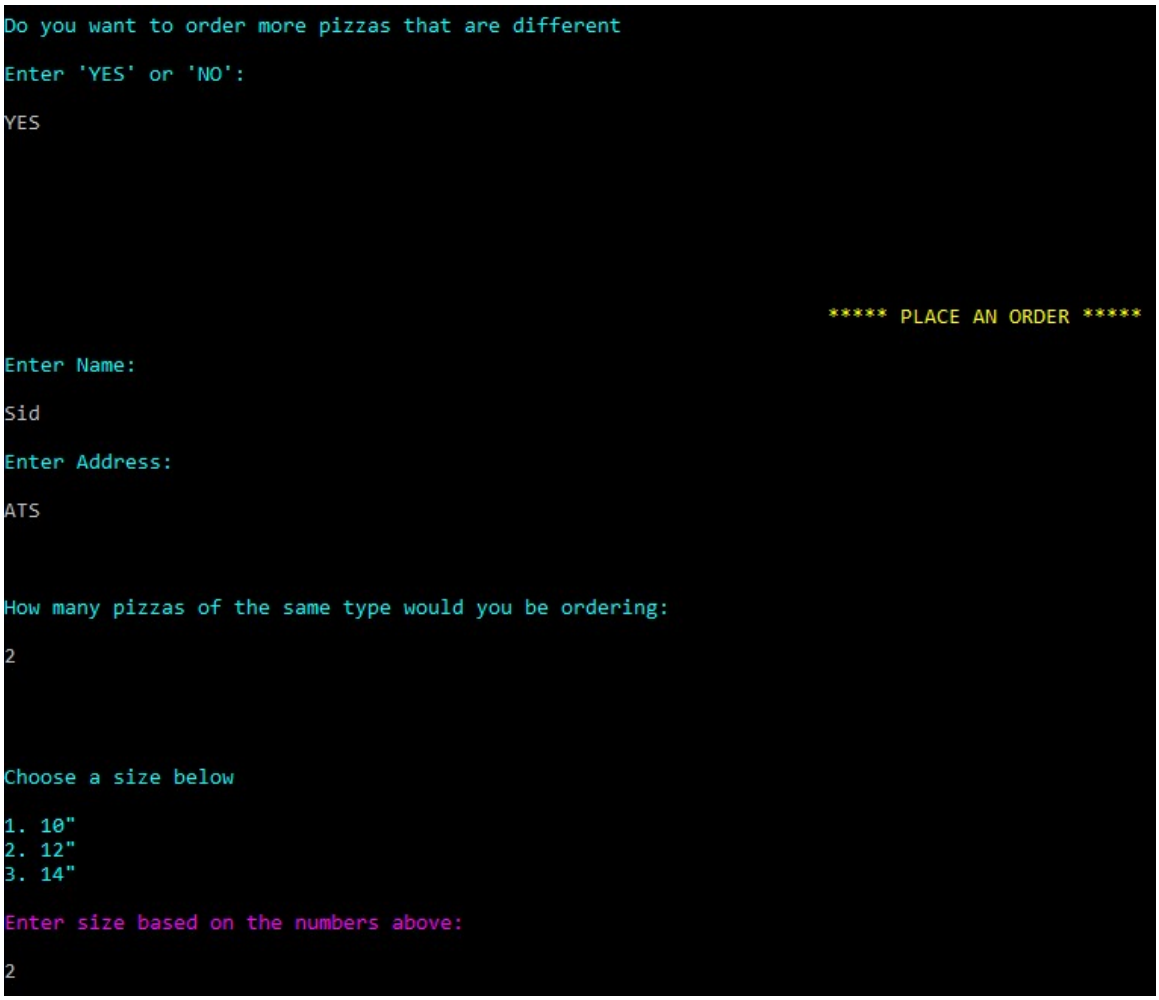
```
      ***** WELCOME TO J.J.J. PIZZA SYSTEM (ALPHA VERSION 2.0) *****

                              Brought to you by -

                                        | ARYAN JAIN | SIDDHARATHA JAISWAL | MANAV JAIN |
```

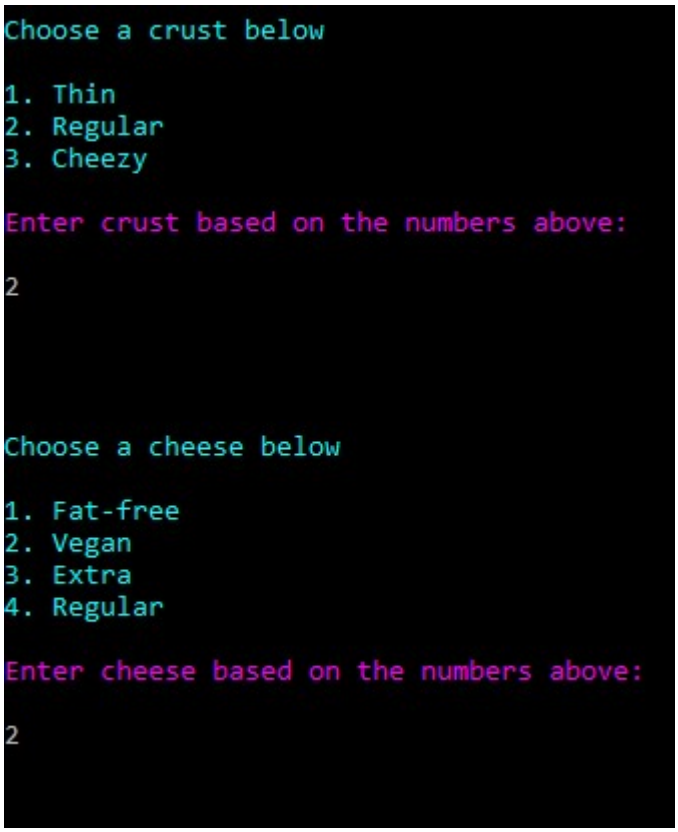
```
Select a command below

1. Place an order
2. View inventory and modify menu

Enter the function based on the numbers above:

1
```

```
                              ***** PLACE AN ORDER *****
```


Enter Name:
Aryan
Enter Address:
ATS
How many pizzas of the same type would you be ordering:
5
Choose a size below
1. 10"
2. 12"
3. 14"
Enter size based on the numbers above:
1
Choose a crust below
1. Thin
2. Regular
3. Cheezy
Enter crust based on the numbers above:
1

```
Choose a cheese below

1. Fat-free
2. Vegan
3. Extra
4. Regular

Enter cheese based on the numbers above:

1



 Choose toppings from below: (Upto 3)
 1. Onion
 2. Green Olives
 3. Jalapeno
 4. Red Paprika
 5. Tomato
 6. Corn
 7. Capsicum
 8. Mushroom

Enter toppings based on the numbers above:



Invalid Topping

Enter toppings based on the numbers above:

2

Enter toppings based on the numbers above:

3

Enter toppings based on the numbers above:

4


Your total amount due is: Rs4300.0
```


Do you want to order more pizzas that are different
Enter 'YES' or 'NO':
YES
***** PLACE AN ORDER *****
Enter Name:
Sid
Enter Address:
ATS
How many pizzas of the same type would you be ordering:
2
Choose a size below
1. 10"
2. 12"
3. 14"
Enter size based on the numbers above:
2


Choose a crust below
1. Thin
2. Regular
3. Cheezy
Enter crust based on the numbers above:
2
Choose a cheese below
1. Fat-free
2. Vegan
3. Extra
4. Regular
Enter cheese based on the numbers above:
2

```
Choose toppings from below: (Upto 3)
1. Onion
2. Green Olives
3. Jalapeno
4. Red Paprika
5. Tomato
6. Corn
7. Capsicum
8. Mushroom

Enter toppings based on the numbers above:

5

Enter toppings based on the numbers above:

6

Enter toppings based on the numbers above:

7


Your total amount due is: Rs481.0



Do you want to order more pizzas that are different

Enter 'YES' or 'NO':

NO


Your order will be deliverd to you within 30 minutes at 2020-02-02 20:52:00

Have a good day.
Thank you.
```

```
                    ***** WELCOME TO J.J.J. PIZZA SYSTEM (ALPHA VERSION 2.0) *****

                                                Brought to you by -

                                                | ARYAN JAIN | SIDDHARATHA JAISWAL | MANAV JAIN |


Select a command below

1. Place an order
2. View inventory and modify menu

Enter the function based on the numbers above:

2
```

```
                                                    ***** VIEW INVENTORY & MODIFY MENU *****


-------------------------------------------------------------------------
|  ID |        INGREDIENT        | QUANTITY | COST PRICE | SELL PRICE  |
-------------------------------------------------------------------------
| 1   | 10" - Thin               | 90       | 60         | 69          |
| 2   | 10" - Regular            | 95       | 50         | 57.5        |
| 3   | 10" - Cheezy             | 100      | 70         | 80.5        |
| 4   | 12" - Thin               | 100      | 70         | 69          |
| 5   | 12" - Regular            | 98       | 60         | 57.5        |
| 6   | 12" - Cheezy             | 100      | 80         | 80.5        |
| 7   | 14" - Thin               | 100      | 80         | 69          |
| 8   | 14" - Regular            | 100      | 70         | 57.5        |
| 9   | 14" - Cheezy             | 100      | 90         | 80.5        |
| 10  | Cheese -> Fat - Free     | 90       | 50         | 57.5        |
| 11  | Cheese -> Vegan          | 98       | 40         | 46          |
| 12  | Cheese -> Extra          | 95       | 30         | 34.5        |
| 13  | Cheese -> Regular        | 100      | 20         | 23          |
| 14  | Onion                    | 100      | 10         | 11.5        |
| 15  | Green Olives             | 95       | 10         | 11.5        |
| 16  | Jalapeno                 | 95       | 15         | 17.25       |
| 17  | Red Paprika              | 90       | 15         | 17.25       |
| 18  | Tomato                   | 88       | 5          | 5.75        |
| 19  | Corn                     | 88       | 5          | 5.75        |
| 20  | Capsicum                 | 93       | 5          | 5.75        |
| 21  | Mushroom                 | 100      | 10         | 11.5        |
-------------------------------------------------------------------------


Would you like to modify the menu

Enter 'YES' or 'NO':

YES

Please enter the ingredient you would like to add:

Green Olives
```

```
Please enter the new cost price of the ingredient:

15



Please enter the new selling price of the ingredient:

17.5



Please enter the name of the ingredient you would like to replace:

Olives
```

ID	INGREDIENT	QUANTITY	COST PRICE	SELL PRICE
1	10" - Thin	90	60	69
2	10" - Regular	95	50	57.5
3	10" - Cheezy	100	70	80.5
4	12" - Thin	100	70	69
5	12" - Regular	98	60	57.5
6	12" - Cheezy	100	80	80.5
7	14" - Thin	100	80	69
8	14" - Regular	100	70	57.5
9	14" - Cheezy	100	90	80.5
10	Cheese -> Fat - Free	90	50	57.5
11	Cheese -> Vegan	98	40	46
12	Cheese -> Extra	95	30	34.5
13	Cheese -> Regular	100	20	23
14	Onion	100	10	11.5
15	Green Olives	95	10	11.5
16	Jalapeno	95	15	17.25
17	Red Paprika	90	15	17.25
18	Tomato	88	5	5.75
19	Corn	88	5	5.75
20	Capsicum	93	5	5.75
21	Mushroom	100	10	11.5

```
Would you like to modify the menu

Enter 'YES' or 'NO':

NO


Have a good day.
Thank you.
```

THANK YOU