

BHAVAN'S RAJAJI VIDYASHRAM

SAS MOBILE SELLER



**BY:
D SANJEEV 12 D**

AIM

The aim of the project is to design and to develop a Python program that uses tables from SQL database that aids in online mobile selling. It provides the user with the following options:

1. To buy a smartphone
2. To buy an accessory
3. To edit databases as admin

INTRODUCTION

The database designed helps in online mobile selling. It includes various tables for dynamic storage and retrieval of data stored in a systemic format. This helps regulate huge amount of data storage and digitalize previous hard-copy versions into a table format. These tables store varying information and also provide options to comb through the data based on the user's preference.

MODULES USED

1.mysql.connector():

It is a library that connects MySql databases from within python and alters back-end values in SQL through code written in python. It also displays the output in python while storing the information in SQL.

2.random():

The Python Random module is a **built-in module** for generating random integers in Python. These numbers occur randomly and does not follow any rules or instructions. We can therefore use this module to generate random numbers, display a random item for a list or string, and so on

FUNCTIONS USED

1. sc():

To display customer details.

2. admin():

To update, delete, modify, mobile and accessories details and also to show report of the performance of the mobile seller and ask for customer feedback.

3. mobile_purchase():

To buy mobile phones from list of smart phones available by the mobile seller.

4. accessory_purchase():

To buy accessories from list of accessories available by the mobile seller.

5. buy():

To ask for customer details and payment method and display bill generated after purchase.

6. print_decorative_banner():

To display main menu of the program.

7. add_accessory_details():

To add accessory details by admin.

8. add_smartphone_details():

To add smartphone details by admin.

9. main():

To display choices to the customer.

DATABASE
FILES
TABLES
USED

```

mysql> select * from smartphones;
+---+---+---+---+---+---+---+---+---+
| sno | name          | brand    | ram   | camera | storage | color   | os     | price  |
+---+---+---+---+---+---+---+---+---+
| 1  | F15            | oppo     | 64    | 32    | 128    | blue    | colorOS | 13    | 50000  |
| 2  | T2              | vivo     | 128   | 32    | 256    | black   | android | 12    | 75000  |
| 3  | IPHONE 14 PRO MAX | apple    | 128   | 32    | 256    | yellow  | iOS    |        | 250000 |
| 4  | NOTE 12          | one plus | 32    | 16    | 32    | grey    | android | 12    | 75000  |
| 5  | iPHONE 14          | apple    | 64    | 32    | 256    | orange  | iOS    |        | 250000 |
| 6  | NOTE 12          | redmi   | 64    | 32    | 256    | green   | android | 12    | 75000  |
| 7  | V27 pro          | vivo     | 64    | 32    | 128    | half white | android | 12    | 75000  |
| 8  | 13 pro           | apple    | 64    | 32    | 256    | red    | iOS    |        | 250000 |
| 9  | 13 pro max       | apple    | 64    | 32    | 256    | grey    | iOS    |        | 250000 |
| 10 | 13               | apple    | 64    | 32    | 256    | green   | iOS    |        | 250000 |
| 11 | nord CE 3         | one plus | 64    | 16    | 128    | gold   | android | 11    | 75000  |
| 12 | A31              | oppo     | 64    | 16    | 128    | silver  | colorOS | 13    | 50000  |
| 13 | NOTE 12 PRO        | redmi   | 32    | 8     | 64    | silver  | android | 12    | 75000  |
| 14 | y73              | mi       | 32    | 8     | 64    | silver  | android |        | 75000  |
| 15 | y93              | mi       | 32    | 16    | 64    | brown   | android |        | 75000  |
| 16 | galaxy Z           | samsung  | 32    | 16    | 64    | orange  | android |        | 100000 |
| 17 | 14 pro max        | apple    | 128   | 64    | 256    | orange  | iOS    |        | 250000 |
| 18 | G11              | nokia   | 128   | 64    | 256    | white   | android |        | 75000  |
| 19 | 2.1              | nokia   | 128   | 64    | 256    | violet  | android |        | 75000  |
| 20 | 11R              | one plus | 64    | 32    | 128    | violet  | android | 11    | 75000  |
| 21 | NORD CE2           | one plus | 64    | 16    | 128    | blue black | android | 11    | 75000  |
| 22 | nord CE3           | one plus | 32    | 16    | 64    | blue black | android | 11    | 75000  |
| 23 | 12 pro             | apple    | 32    | 16    | 64    | red blue | iOS    |        | 250000 |
| 24 | 12 pro max         | apple    | 32    | 16    | 128    | half white | iOS    |        | 250000 |
| 25 | 11i               | xiaomi  | 64    | 32    | 128    | half white | android |        | 75000  |
| 26 | galaxy A71          | samsung  | 64    | 32    | 128    | black   | android |        | 100000 |
| 27 | V27                | vivo     | 64    | 32    | 128    | black   | android | 12    | 75000  |
| 28 | A15              | oppo     | 128   | 64    | 256    | blue    | colorOS | 13    | 50000  |
| 30 | ROG 7 ++           | Asus     | 16    | 24    | 256    | White   | Android | 14    | 120000 |
+---+---+---+---+---+---+---+---+---+
29 rows in set (0.01 sec)

```

```

mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| customer_id | int        | NO   | PRI | NULL    | auto_increment |
| customer_name | varchar(255) | YES  |     | NULL    |                |
| phone_number | varchar(20)  | YES  |     | NULL    |                |
| payment_method | varchar(50) | YES  |     | NULL    |                |
| item_name    | varchar(255) | YES  |     | NULL    |                |
| discount      | int        | YES  |     | NULL    |                |
| price         | float      | YES  |     | NULL    |                |
| date          | date       | YES  |     | 2023-01-01 |                |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)

```

```
mysql> select * from accessories;
+-----+-----+-----+-----+-----+-----+-----+
| serialno | ProductName | Brand | ModelNo | ProductId | Cost |
+-----+-----+-----+-----+-----+-----+
| 1 | Phone Case | Army | Tough Armor | 23 | 2500 |
| 2 | Bluetooth Speaker | Boat | MEGABOOM 3 | 129 | 1800 |
| 3 | Apple Pro Vision | Apple | Foreign Product | 142 | 290000 |
| 4 | AirPods | Apple | Foreign Product | 143 | 45000 |
| 5 | Screen Protector | GorillaTemper | Glass Screen Protector | 145 | 1500 |
| 6 | Gaming Mouse | Asus | M65 RGB Elite | 147 | 6000 |
| 7 | USB-C Cable | SamSung | PowerLine+ | 156 | 1000 |
| 8 | Wireless Earphones | Boat | Elite 75t | 166 | 1500 |
| 9 | Car Phone Mount | Audi | Easy One Touch 5 | 168 | 2000 |
| 10 | Wireless Charging Pad | Nothing | Boost Charge | 212 | 4000 |
| 11 | Mechanical Keyboard | Asus | Apex Pro | 213 | 20000 |
| 12 | Fitness Tracker | MicroMax | Charge 4 | 214 | 13000 |
| 13 | Camera Bag | Canva | ProTactic BP 350 AW II | 220 | 1200 |
| 14 | Smart Home Hub | Google | Nest Hub | 231 | 8000 |
| 15 | Power Bank | SamSung | JP-101 | 314 | 2500 |
| 16 | USB C | SAMSUNG | T-150 | 56 | 550 |
| 17 | case | oppo | 78 | 88 | 7800 |
+-----+-----+-----+-----+-----+-----+
17 rows in set (0.02 sec)
```

```
mysql> select * from admin;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 1 | san | 56 |
| 2 | mj | 123 |
| 3 | sus | 36 |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

PROJECT

CODE

```

import mysql.connector as ms
import random as r

Con = ms.connect(host="localhost", user="root", password="sql@123",
database="online_mobile_seller")
MCursor = Con.cursor()

def cus_week():
    MCursor.execute("SELECT customer_name, SUM(price) AS total_purchase, count(*)
    FROM customer
    WHERE YEARWEEK(date) = YEARWEEK(NOW())
    GROUP BY customer_name
    ORDER BY Count(*) DESC
    LIMIT 1")
    F=MCursor.fetchone()
    if F is not None:
        ol=F[0]
        return ol
    else:
        return None

def sc():
    MCursor.execute("SELECT * FROM customer")
    customers = MCursor.fetchall()
    print("Serial Date Customer Name Phone Number Payment Type Item
Name Discount Final Amount")
    print("-----")
    for customer in customers:
        customer_id, customer_name, phone_number, payment_method, item_name,
        discount, final_amount, date = customer
        print(f'{customer_id:<8} {str(date):<15} {customer_name:<25}
{phone_number:<18} {payment_method:<23} {item_name:<25} {discount:<11}
{final_amount:<15}')


def admin():
    MCursor.execute('SELECT * FROM Admin')
    User = MCursor.fetchall()
    max_tries = 3

    for attempt in range(max_tries):
        u = input("Enter username: ").lower()
        p = int(input("Enter password: "))

        login_successful = False

        for i in User:
            if u == i[1] and p == i[2]:
                print('Login successful!')

```

```

login_successful = True
break
if login_successful:
    break
else:
    remaining_tries = max_tries - (attempt + 1)
    print(f'Invalid username or password. Tries remaining: {remaining_tries}')
    print('Or IF YOU WANT TO QUIT TYPE "1"')
    po=input()
    if po == '1':
        main()

if remaining_tries == 0:
    print('Maximum number of tries reached. Exiting..')
    break

if login_successful:
    while True:
        if i[1] == 'san':
            print('Welcome Admin SanjeevD\nWhat Do You Want To Modify:')
        elif i[1] == 'mj':
            print('Welcome Admin ANSHUKMAN MJ\nWhat Do You Want To Modify:')
        elif i[1] == 'sus':
            print('Welcome Admin CNC SUSINDHARAN\nWhat Do You Want To Modify:')

        print('1) SmartPhone\n2) Accessories\n3) ShowReport\n4) Quit')
        a = int(input())

        if a == 1:
            print('Do You Want\n1) Update\n2) Delete\n3) Add Records\n4) Quit')
            b = int(input())
            if b == 1:
                MCursor.execute("SELECT * FROM smartphones")
                mobiles = MCcursor.fetchall()
                print("Serial Mobile Name      Brand          RAM  Camera Storage")
                print("Color                  OS   Price")
                print("-----")
                for mobile in mobiles:
                    print(f'{mobile[0]}:{<6} {mobile[1]}:{<20} {mobile[2]}:{<20} {mobile[3]}:{<6} {mobile[4]}:{<6} {mobile[5]}:{<8} {mobile[6]}:{<10} {mobile[7]}:{<15} {mobile[8]}:{<6}')
                print('S.no you want to change')
                s = int(input())
                if s== 0 :
                    break
                MCcursor.execute('select * from smartphones where sno = %s', (s,))
                op = MCcursor.fetchall()
                if not op:

```

```

print("Record not found.")
break

for z in op:
    print('Enter what you want to
change:\n1)name\n2)brand\n3)ram\n4)camera\n5)storage\n6)colour\n7)os\n
8)price\n9)Quit')
    qe = int(input())
    if qe == 1:
        name = input("Enter new name: ")
        MCursor.execute('update smartphones set name = %s where sno = %s',
        (name, s))
        print('Update Success')
        Con.commit()
        break
    elif qe == 2:
        brand = input("Enter new brand: ")
        MCursor.execute('update smartphones set brand = %s where sno =
        %s', (brand, s))
        print('Update Success')
        Con.commit()
        break
    elif qe == 3:
        ram = int(input("Enter new RAM: "))
        MCursor.execute('update smartphones set ram = %s where sno = %s',
        (ram, s))
        print('Update Success') Con.commit()
        break
    elif qe == 4:
        camera = int(input("Enter new camera: "))
        MCursor.execute('update smartphones set camera = %s where sno =
        %s', (camera, s))
        print('Update Success') Con.commit()
        break
    elif qe == 5:
        storage = int(input("Enter new storage: "))
        MCursor.execute('update smartphones set storage = %s where sno =
        %s', (storage, s))

        print('Update Success')
        Con.commit()
        break
    elif qe == 6:
        colour = input("Enter new colour: ")
        MCursor.execute('update smartphones set color = %s where sno = %s',
        (colour, s))

        print('Update Success')
        Con.commit()
        break
    elif qe == 7:

```

```

os = input("Enter new OS: ")
MCursor.execute('update smartphones set os = %s where sno = %s',
(os, s))

print('Update Success')
Con.commit()
break
elif qe == 8:
    price = int(input("Enter new price: "))
    MCursor.execute('update smartphones set price = %s where sno = %s',
(price, s))
    print('Update Success')
    Con.commit()
    break
else:
    break
elif b == 2:
    MCursor.execute("SELECT * FROM smartphones")
    mobiles = MCursor.fetchall()
    print("Serial Mobile Name      Brand          RAM  Camera Storage Color
OS  Price")
    print("-----")
    for mobile in mobiles:
        print(f'{mobile[0]:<6} {mobile[1]:<20} {mobile[2]:<20} {mobile[3]:<6}
{mobile[4]:<6} {mobile[5]:<8} {mobile[6]:<10} {mobile[7]:<15}
{mobile[8]:<6}') print('S.no you want to DELETE')
        if sno == 0:
            break
    MCursor.execute('delete from smartphones where sno = %s', (sno,))
    print('Successfully Deleted Record Thank You')
    Con.commit()
    break
elif b == 3:
    add_smartphone_details()
    break
elif b == 4:
    break
else:
    print('Invalid option')
elif a == 2:
    print('Do You Want\n1) Update\n2) Delete\n3) Add Records\n4) Quit')
    b = int(input())
    if b == 1:
        while True:
            MCursor.execute("SELECT * FROM accessories")
            accessories = MCursor.fetchall()
            print("Serial Product Name      Brand  Model No      Product ID
Cost")
            print("-----")

```

```

for accessory in accessories:
    print(f'{accessory[0]:<6} {accessory[1]:<27} {accessory[2]:<7}
          {accessory[3]:<20} {accessory[4]:<10}
          {accessory[5]:<4}') print('Serial No
you want to change') serial_no =
int(input())
if serial_no == 0:
    break
MCursor.execute('SELECT * FROM accessories WHERE serialno = %s',
(serial_no,))

accessory = MCursor.fetchone()

if accessory:
    print('Enter what you want to change:')
    print('1) ProductName\n2) Brand\n3) ModelNo\n4) ProductId\n5) Cost')
    f = int(input())

    if f == 1:
        new_value = input("Enter new ProductName: ")
        MCursor.execute('UPDATE accessories SET ProductName= %s
WHERE serialno = %s', (new_value, serial_no))
        print('Update successful!')
        Con.commit()
    elif f == 2:
        new_value = input("Enter new Brand: ")
        MCursor.execute('UPDATE accessories SET Brand= %s WHERE
serialno = %s', (new_value, serial_no))
        print('Update successful!')
        Con.commit()
    elif f == 3:
        new_value = input("Enter new ModelNo: ")
        MCursor.execute('UPDATE accessories SET ModelNo= %s WHERE
serialno = %s', (new_value, serial_no))
        print('Update successful!')
        Con.commit()
    elif f == 4:
        new_value = int(input("Enter new ProductId: "))
        MCursor.execute('UPDATE accessories SET Productid= %s WHERE
serialno = %s', (new_value, serial_no))
        print('Update successful!')
        Con.commit()
    elif f == 5:
        new_value = int(input("Enter new Cost: "))
        MCursor.execute('UPDATE accessories SET Cost= %s WHERE
serialno = %s', (new_value, serial_no))
        print('Update successful!')
        Con.commit()
    else:
        print("Invalid option")
        return

```

```

    elif b == 2:
        while True:
            MCursor.execute("SELECT * FROM accessories")
            accessories = MCursor.fetchall()
            print("Serial Product Name      Brand Model No      Product ID
                  Cost")
            print("-----")
            for accessory in accessories:
                print(f'{accessory[0]:<6} {accessory[1]:<27} {accessory[2]:<7}
{accessory[3]:<20} {accessory[4]:<10}
{accessory[5]:<4}") print('Serial No you
want to DELETE') serial_no = int(input())
            if serial_no == 0:
                break
            else:
                MCursor.execute('DELETE FROM accessories WHERE serialno = %s',
                               (serial_no,))

                print('Delete successful!')
                Con.commit()
        elif b == 3:
            add_accessory_details()
            break
        elif b == 4:
            break
        else:
            print('Invalid Try Again')
            print('Accessory not found.')

    elif a == 3:
        def max_purchase_amount_by_customer():
            MCursor.execute("""SELECT customer_id, customer_name, MAX(price) AS
max_purchase_amount
FROM customer
GROUP BY customer_id, customer_name
ORDER BY max_purchase_amount DESC
LIMIT 2;
""")
            r=MCursor.fetchall()
            for i in range (0,2):
                print(i+1,'Max amount purchase is')
                print(r[i][2])
                print('The Customer gave most amount is\n',r[i][1].upper())
        def frequent_customer():
            MCursor.execute("""
SELECT MIN(customer_id) AS customer_id, customer_name,
COUNT(*) AS purchase_count
FROM customer
GROUP BY customer_name
ORDER BY purchase_count DESC
""")

```

```

        LIMIT 2; """")
r=MCursor.fetchall()
for i in range (0,2):
    print('The',i+1,'Star Customer Is:','With Purchase count',r[i][2])
    print(r[i][1].upper())
def no_purchase_of_payment_method():
    MCursor.execute("""
    SELECT payment_method, COUNT(*) AS payment_count
    FROM customer
    GROUP BY payment_method;""")
s=MCursor.fetchall()
for i in s:
    print('The Payment\'s done by method\n',i[0],'Is:',i[1])
def total_amount_got():
    MCursor.execute("""
    SELECT SUM(price) AS Amount_Given_By_All_Customers
    FROM customer""")
s=MCursor.fetchall()
for i in s:
    print("Amount_Given_By_All_Customers:\n",i)
def montly_amount_recieved():
    query = """
    SELECT YEAR(date) AS year, MONTH(date) AS month, SUM(price) AS
    total_amount
    FROM customer
    GROUP BY YEAR(date), MONTH(date) """
    MCursor.execute(query)
    result = MCursor.fetchall()
    for row in result:
        print(f"Year: {row[0]}, Month: {row[1]}, Total Amount Recieved:
{row[2]}")
def lascus():
    MCursor.execute("SELECT * FROM customer ORDER BY customer_id
DESC LIMIT 1")
    last_customers = MCursor.fetchone()
    if last_customers:
        last_customer = last_customers
        print("{:<8} {:<15} {:<25} {:<15} {:<20} {:<15} {:<11}
{:<15}".format(
            "Serial", "Date", "Customer Name", "Phone Number", "Payment
Type", "Item Name", "Discount", "Final Amount"))
        print("*" * 125)
        print("{:<8} {:<15} {:<25} {:<15} {:<20} {:<25} {:<12}
{:<15}".format(
            last_customer[0], str(last_customer[7]), last_customer[1],
            last_customer[2], last_customer[3], last_customer[4],
            last_customer[5], last_customer[6])) else:
            print("No customer found.")
def Billww():
    print('Date Format = 20xx-0x(month)-xx(date)')

```

```

date = input('Enter Starting date You want To Search: ')
datee = input("Enter Ending date You want to Search: ")

MCursor.execute('SELECT * FROM customer WHERE date
BETWEEN %s AND %s', (date, datee))
customers = MCursor.fetchall()
while True:
    if customers:
        print("{:<8} {:<15} {:<25} {:<15} {:<20} {:<15} {:<11}
{:<15}".format(
            "Serial", "Date", "Customer Name", "Phone Number", "Payment
Type", "Item Name", "Discount", "Final Amount"))
        print("-" * 105)

        for last_customer in customers:
            print("{:<8} {:<15} {:<25} {:<15} {:<20} {:<15} {:<11}
{:<15}".format(last_customer[0], str(last_customer[7]),
last_customer[1],
last_customer[2], last_customer[3], last_customer[4],
last_customer[5], last_customer[6])) break
    else:
        print('NO CUSTOMER BETWEEN THAT DATE RANGE')
        if input('Or If You Want To Quit Press Q').lower() == 'q':
            break

print("The Reports You Want To Show:\n1 - Two Star \Customer\n2 -
Paymentmethod Total\n3 - Max Amount Of Money Spend by first 2
customers\n4 - Total Amount Got\n5 - MONTHLY AMOUNT RECIEVED
\n6 - SHOWCUSTOMERS\n7 - SHOWFEEDBACK\n8 - Shows Last customer\n9 - Bills within
Two Date Interval\n10 - Quit")
pp=int(input())
while True:
    if pp == 1:
        frequent_customer()
        break
    elif pp == 2:
        no_purchase_of_payment_method()
        break
    elif pp == 3:
        max_purchase_amount_by_customer()
        break
    elif pp == 4:
        total_amount_got()
        break
    elif pp == 5:
        montly_amount_recieved()
        break
    elif pp == 6:
        sc()

```

```

        break
    elif pp == 7:
        feedback_file = "feedback.txt"
        try:
            with open(feedback_file, "r") as f:
                contents = f.read()
                print(contents)
                break
        except FileNotFoundError:
            print("The feedback file does not exist.")
    elif pp == 8:
        lascus()
        break
    elif pp == 9:
        Billww()
        break
    elif pp == 10:
        break
    else:
        print('Invalid and Try Again')
elif a == 4:
    break
else:
    print('Invalid option')

```

```

def mobile_purchase():
    MCursor.execute("SELECT * FROM smartphones")
    mobiles = MCursor.fetchall()
    print("Serial Mobile Name      Brand          RAM  Camera Storage Color   OS
Price")
    print("-----")
    for mobile in mobiles:
        print(f'{mobile[0]:<6} {mobile[1]:<20} {mobile[2]:<20} {mobile[3]:<6}
{mobile[4]:<6} {mobile[5]:<8} {mobile[6]:<10}{mobile[7]:<15}{mobile[8]:<6}')

    serial_number = int(input("Enter Serial Number of the mobile you want to buy: "))
    MCcursor.execute("SELECT * FROM smartphones WHERE sno = %s", (serial_number,))
    selected_mobile = MCcursor.fetchone()

    if selected_mobile:
        print(f"Your desired mobile is {selected_mobile[1]}. Its details are as follows:")
        print(f"Serial Number: {selected_mobile[0]}")
        print(f"Name: {selected_mobile[1]}")
        print(f"Brand: {selected_mobile[2]}")
        print(f"RAM: {selected_mobile[3]}")
        print(f"Camera: {selected_mobile[4]}")

```

```

print(f"Storage: {selected_mobile[5]}")
print(f"Color: {selected_mobile[6]}")
print(f"OS: {selected_mobile[7]}")
print(f"Cost: {selected_mobile[8]}")

ask_buy = input("Do you want to buy it? (Y - YES/N - NO): ").strip().upper()
if ask_buy == "Y":
    buy(selected_mobile[1], selected_mobile[8])
else:
    print("Visit Again")
else:
    print("Mobile not found. Please try again.")

def accessory_purchase():
    MCursor.execute("SELECT * FROM accessories")
    accessories = MCursor.fetchall()
    print("Serial Product Name      Brand Model No      Product ID Cost")
    print("-----")
    for accessory in accessories:
        print(f" {accessory[0]:<6} {accessory[1]:<27} {accessory[2]:<7} {accessory[3]:<20} "
              f" {accessory[4]:<10} {accessory[5]:<4}")

    serial_number = int(input("Enter Serial Number of the accessory you want to buy: "))
    MCursor.execute("SELECT * FROM accessories WHERE serialno = %s",
                   (serial_number,))
    selected_accessory = MCursor.fetchone()

    if selected_accessory:
        print(f"Your desired accessory is {selected_accessory[1]}. Its details are as follows:")
        print(f"Serial Number: {selected_accessory[0]}")
        print(f"Product Name: {selected_accessory[1]}")
        print(f"Brand: {selected_accessory[2]}")
        print(f"Model No: {selected_accessory[3]}")
        print(f"Product ID: {selected_accessory[4]}")
        print(f"Cost: {selected_accessory[5]}")

        ask_buy = input("Do you want to buy it? (Y - YES/N - NO): ").strip().upper()
        if ask_buy == "Y":
            buy(selected_accessory[1], selected_accessory[5])
        else:
            print("Visit Again")
    else:
        print("Accessory not found. Please try again.")

def buy(item_name, Price):
    while True:
        phone_number = input("Enter phone number: ")
        if len(phone_number) == 10 and phone_number.isdigit():

```

```

        phone_number = '+91' + phone_number
        break
    else:
        print('Invalid phone number. Please enter a 10-digit number.')
        print('Or Do You want to Quit Press 1 \nTo Continue Enter 0')
        po=input()
        if po == '1':
            main()

customer_name = input("Enter customer name: ")

print(' Do You want to Quit Press 1 or \nTo Continue Enter 0')
if input() == '1':
    main()

print('Your Mode Of Payment')
print('1 - UPI')
print('2 - Cash')
print('3 - Card')
print('Or To Quit Press 0')
while True:
    ch = input()
    if ch == '1':
        payment_type = 'UPI'
        break
    elif ch == 0:
        main()
    elif ch == '2':
        payment_type = 'Cash'
        break
    elif ch == '3':
        p = 'Card'
        if p == 'Card':
            print('1 - Credit Card')
            print('2 - Debit Card')
            ok = input()
            while True:
                if ok == '1':
                    payment_type = 'Card - Credit'
                    break
                else:
                    payment_type = 'Card - Debit'
                    break
            break
        else:
            print('Not available')
d = r.randint(0, 4)
MCursor.execute("SELECT customer_name, phone_number,COUNT(*) as purchase_count

```

```

FROM customer
GROUP BY customer_name, phone_number")
nb=MCursor.fetchall()
p=cus_week()
c =
for i in nb:
    if i[0] == customer_name and i[1] == phone_number :
        d = d+(i[2]/2)
        print(i[2])
        print(i[2]/2)
    if i[0] == p and p == customer_name :
        d=d+5
        c = i[0]
        break
if d >= 25 and c == p:
    d=0
elif d >=15 and c!=p:
    d=0

afterdiscount=Price-(Price*(d/100))
kl = "SELECT curdate()"
MCursor.execute(kl)
t = MCursor.fetchone()
MCursor.execute("INSERT INTO customer (customer_name, phone_number,
payment_method, item_name, discount, price , date) VALUES (%s, %s, %s, %s, %s, %s,
%s)",(customer_name, phone_number, payment_type, item_name, d, afterdiscount,t[0]))
Con.commit()
MCursor.execute('SELECT * FROM customer')
customersss = MCursor.fetchall()

if customersss:
    last_customer=customersss[-1]
    print('Here Is Your Bill')
    print("{:<8} {:<15} {:<25} {:<15} {:<20} {:<15} {:<11} {:<15}".format(
        "Serial", "Date", "Customer Name", "Phone Number", "Payment Type", "Item
        Name", "Discount", "Final Amount"))
    print("*" * 110)
    print("{:<8} {:<15} {:<25} {:<15} {:<20} {:<15} {:<11} {:<15}".format(
        last_customer[0], str(last_customer[7]), last_customer[1], last_customer[2],
        last_customer[3], last_customer[4], last_customer[5], last_customer[6]))
else:
    print("No customer found.")

print("Payment done.")
print("Thank you for your purchase!\n Do You want to Give Feedback")
ph=input('Y - Yes / N - No').upper()

```

```

if ph == 'Y':
    feedback_file = "feedback.txt"
    user_feedback = input("Enter your feedback: ")
    with open(feedback_file, "a") as f:
        f.write(f'Thank You Customer For FeedBack {customer_name}\n')
        while True:
            p=int(input('Rate us Out Of 5'))
            if p>5:
                print('Try Again')
            else:
                break
        f.write(f'Our Rating By {customer_name} is {p} Out of 5\n')
        f.write(user_feedback + "\n")
        print('if you give any more feedback please\n continue by pressing 1 else press 0')
        sss=int(input())
        if sss == 1:
            user_feedback = input("Enter your feedback: ")
            f.write(user_feedback + "\n")
Con.commit()

```

```

# MAIN FUNCTION STARTS FROM HERE !!!
def print_decorative_banner(text):
    decorative_line = "█" * (len(text) + 4)
    print(decorative_line)
    print(f"█ {text} █")
    print("█ By Dev - Anshukman MJ , Sanjeev D , Susindharen CNC █")
    print(decorative_line)

def add_accessory_details():
    product_name = input("Enter product name: ")
    brand = input("Enter brand: ")
    model_no = input("Enter model number: ")
    product_id = input("Enter product ID: ")
    cost = float(input("Enter cost: "))

    MCursor.execute("INSERT INTO accessories (productname, brand, modelno, productid, cost) VALUES (%s, %s, %s, %s, %s)",
                   (product_name, brand, model_no, product_id, cost))
    Con.commit()
    print("Accessory details added successfully!")

```

```

def add_smartphone_details():
    mobile_name = input("Enter mobile name: ")
    brand = input("Enter brand: ")
    ram = input("Enter RAM: ")
    camera = input("Enter camera: ")
    storage = input("Enter storage: ")
    color = input("Enter color: ")
    os = input("Enter OS: ")
    price = float(input("Enter price: "))

    MCursor.execute("INSERT INTO smartphones (name, brand, ram, camera, storage, color, os, price) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)", (mobile_name, brand, ram, camera, storage, color, os, price))
    Con.commit()
    print("Smartphone details added successfully!")

def main():
    while True:
        p=cus_week()
        print_decorative_banner("          Welcome to Online Mobile Seller!      ")
        print('OUR CUSTOMER OF THE WEEK IS::',end="")
        print(f"\n {p}\n")
        print('(1)M - MobilePurchase\n(2)A - AccessoryPurchase\n(3)Q - Quit\n(4)U - Admin')
        choice = input("CHOOSE THE OPTION YOU WANT      ").upper()
        if choice == "M" or choice == '1':
            mobile_purchase()
        elif choice == "A" or choice == '2':
            accessory_purchase()
        elif choice == "Q" or choice == '3':
            break
        elif choice == 'U' or choice == '4':
            admin()
        else:
            print("Invalid choice. Please try again.")

    if __name__ == "__main__":
        main()

MCursor.close()
Con.close()

```

OUTPUTS

✿ Welcome to Online Mobile Seller! ✿
By Dev - Anshukman MJ✿, Sanjeev D✿, Susindharen CNC✿

OUR CUSTOMER OF THE WEEK IS:: poi

- (1)M - MobilePurchase
- (2)A - AccessoryPurchase
- (3)Q - Quit
- (4)U - Admin

CHOOSE THE OPTION YOU WANT ✿ M

Serial Number	Brand	RAM	Camera	Storage	Color	OS	Price	
1	F15	oppo	64	32	128	blue	colorOS 13	50000
2	T2	vivo	128	32	256	black	android 12	75000
3	IPHONE 14 PRO MAX	apple	128	32	256	yellow	iOS	250000
4	NOTE 12	one plus	32	16	32	grey	android 12	75000
5	IPHONE 14	apple	64	32	256	orange	iOS	250000
6	NOTE 12	redmi	64	32	256	green	android 12	75000
7	V27 pro	vivo	64	32	128	half white	android 12	75000
8	13 pro	apple	64	32	256	red	iOS	250000
9	13 pro max	apple	64	32	256	grey	iOS	250000
10	13	apple	64	32	256	green	iOS	250000
11	nord CE 3	one plus	64	16	128	gold	android 11	75000
12	A31	oppo	64	16	128	silver	colorOS 13	50000
13	NOTE 12 PRO	redmi	32	8	64	silver	android 12	75000
14	y73	mi	32	8	64	silver	android	75000
15	y93	mi	32	16	64	brown	android	75000
16	galaxy Z	samsung	32	16	64	orange	android	100000
17	14 pro max	apple	128	64	256	orange	iOS	250000
18	G11	nokia	128	64	256	white	android	75000
19	2.1	nokia	128	64	256	violet	android	75000
20	11R	one plus	64	32	128	violet	android 11	75000
21	NORD CE2	one plus	64	16	128	blue black	android 11	75000
22	nord CE3	one plus	32	16	64	blue black	android 11	75000
23	12 pro	apple	32	16	64	red blue	iOS	250000
24	12 pro max	apple	32	16	128	half white	iOS	250000
25	11i	xiaomi	64	32	128	half white	android	75000
26	galaxy A71	samsung	64	32	128	black	android	100000
27	V27	vivo	64	32	128	black	android 12	75000
28	A15	oppo	128	64	256	blue	colorOS 13	50000
29	ROG 7 ++	Asus	16	24	256	White	Android 14	120000

Enter Serial Number of the mobile you want to buy: |

Activate Windows
Go to Settings to activate Windows.

```
Enter Serial Number of the mobile you want to buy: 1
Your desired mobile is F15. Its details are as follows:
Serial Number: 1
Name: F15
Brand: oppo
RAM: 64
Camera: 32
Storage: 128
Color: blue
OS: colorOS 13
Cost:50000
Do you want to buy it? (Y - YES/N - NO): |
```

```
Do you want to buy it? (Y - YES/N - NO): Y
Enter phone number: 1357893456
Enter customer name: qwrtyuy
Do You want to quit Press 1 or
To Continue Enter 0
0
Your Mode Of Payment
1 - UPI
2 - Cash
3 - Card
Or To Quit Press 0
```

```
2
Here Is Your Bill
Serial Date Customer Name Phone Number Payment Type Item Name Discount Final
Amount
=====
50 2023-11-18 qwrtyuy +911357893456 Cash F15 4 48000
.0
Payment done.
Thank you for your purchase!
```

Do You want to Give Feedback

Y - Yes / N - Noy

Enter your feedback: qtyykuy very good seller

Rate us Out Of 54

if you give any more feedback please
continue by pressing 1 else press 0

0

CHOOSE THE OPTION YOU WANT # A

	Serial Product Name	Brand	Model No	Product ID	Cost
1.	Phone Case	Army	Tough Armor	23	2500
2.	Bluetooth Speaker	Boat	MEGABOOM 3	129	1800
3.	Apple Pro Vision	Apple	Foreign Product	142	290000
4.	AirPods	Apple	Foreign Product	143	45000
5.	Screen Protector	GorillaTemper Glass	Screen Protector	145	1500
6.	Gaming Mouse	Asus	M65 RGB Elite	147	6000
7.	USB-C Cable	Samsung	PowerLine+	156	1000
8.	Wireless Earphones	Boat	Elite 75t	166	1500
9.	Car Phone Mount	Audi	Easy One Touch 5	168	2000
10.	Wireless Charging Pad	Nothing	Boost Charge	212	4000
11.	Mechanical Keyboard	Asus	Apex Pro	213	20000
12.	Fitness Tracker	MicroMax	Charge 4	214	13000
13.	Camera Bag	Canva	ProTactic BP 350 AW II	220	1200
14.	Smart Home Hub	Google	Nest Hub	231	8000
15.	Power Bank	SamSung	JP-101	314	2500
16.	USB C	SAMSUNG	T-150	56	550
17.	case	oppo	78	88	7800

Enter Serial Number of the accessory you want to buy: 1

Your desired accessory is Phone Case. Its details are as follows:

Serial Number: 1

Product Name: Phone Case

Brand: Army

Model No: Tough Armor

Product ID: 23

Cost: 2500

Do you want to buy it? (Y - YES/N - NO): |

Do you want to buy it? (Y - YES/N - NO) : Y
Enter phone number: 2467843225
Enter customer name: WDFGUYR
Do You want to Quit Press 1 or
To Continue Enter 0
0

Your Mode Of Payment

1 - UPI
2 - Cash
3 - Card
Or To Quit Press 0
3
1 - Credit Card
2 - Debit Card
2

Here Is Your Bill

Serial	Date	Customer Name	Phone Number	Payment Type	Item Name	Discount	Final
51	2023-11-18	WDFGUYR	+912467843225	Card - Debit	Phone Case	4	2400.

Payment done.
Thank you for your purchase!

CHOOSE THE OPTION YOU WANT * U
Enter username: MJ
Enter password: 123
Login successful!
Welcome Admin ANSHUKMAN MJ
What Do You Want To Modify:
1) SmartPhone
2) Accessories
3) ShowReport
4) Quit

Enter username: MJ

Enter password: 123

Login successful!

Welcome Admin ANSHUKMAN MJ

What Do You Want To Modify:

- 1) SmartPhone
- 2) Accessories
- 3) ShowReport
- 4) Quit

1

Do You Want

- 1) Update
- 2) Delete
- 3) Add Records
- 4) Quit

|

Serial	Mobile Name	Brand	RAM	Camera	Storage	Color	OS	Price
1	F15	oppo	64	32	128	blue	colorOS 13	50000
2	T2	vivo	128	32	256	black	android 12	75000
3	IPHONE 14 PRO MAX	apple	128	32	256	yellow	ios	250000
4	NOTE 12	one plus	32	16	32	grey	android 12	75000
5	IPHONE 14	apple	64	32	256	orange	ios	250000
6	NOTE 12	redmi	64	32	256	green	android 12	75000
7	V27 pro	vivo	64	32	128	half white	android 12	75000
8	13 pro	apple	64	32	256	red	ios	250000
9	13 pro max	apple	64	32	256	grey	ios	250000
10	13	apple	64	32	256	green	ios	250000
11	nord CE 3	one plus	64	16	128	gold	android 11	75000
12	A11	oppo	64	16	128	silver	colorOS 13	50000
13	NOTE 12 PRO	redmi	32	8	64	silver	android 12	75000
14	y73	mi	32	8	64	silver	android	75000
15	y53	mi	32	16	64	brown	android	75000
16	galaxy Z	samsung	32	16	64	orange	android	100000
17	14 pro max	apple	128	64	256	orange	ios	250000
18	G11	nokia	128	64	256	white	android	75000
19	2.1	nokia	128	64	256	violet	android	75000
20	11R	one plus	64	32	128	violet	android 11	75000
21	NORD CE2	one plus	64	16	128	blue black	android 11	75000
22	nord CE3	one plus	32	16	64	blue black	android 11	75000
23	12 pro	apple	32	16	64	red blue	ios	250000
24	12 pro max	apple	32	16	128	half white	ios	250000
25	11i	xiaomi	64	32	128	half white	android	75000
26	galaxy A71	samsung	64	32	128	black	android	100000

S.no you want to change

2

Enter what you want to change:

- 1)name
- 2)brand
- 3)ram
- 4)camera
- 5)storage
- 6)colour
- 7)os
- 8)price
- 9)Quit

7

Enter new OS: ios

Update Success

S.no you want to DELETE

2

Successfully Deleted Record Thank You

```
S.no you want to change  
30  
Enter what you want to change:  
1)name  
2)brand  
3)ram  
4)camera  
5)storage  
6)colour  
7)os  
8)price  
9)Quit  
9
```

```
Do You Want  
1) Update  
2) Delete  
3) Add Records  
4) Quit  
1  


| Serial | Product Name          | Brand                                | Model No               | Product ID | Cost   |
|--------|-----------------------|--------------------------------------|------------------------|------------|--------|
| 1      | Phone Case            | Army                                 | Tough Armor            | 23         | 2500   |
| 2      | Bluetooth Speaker     | Boat                                 | MEGABOOM 3             | 129        | 1800   |
| 3      | Apple Pro Vision      | Apple                                | Foreign Product        | 142        | 290000 |
| 4      | AirPods               | Apple                                | Foreign Product        | 143        | 45000  |
| 5      | Screen Protector      | GorillaTemper Glass Screen Protector | 145                    |            | 1500   |
| 6      | Gaming Mouse          | Asus                                 | M65 RGB Elite          | 147        | 6000   |
| 7      | USB-C Cable           | SamSung                              | PowerLine+             | 156        | 1000   |
| 8      | Wireless Earphones    | Boat                                 | Elite 75t              | 166        | 1500   |
| 9      | Car Phone Mount       | Audi                                 | Easy One Touch 5       | 168        | 2000   |
| 10     | Wireless Charging Pad | Nothing                              | Boost Charge           | 212        | 4000   |
| 11     | Mechanical Keyboard   | Asus                                 | Apex Pro               | 213        | 20000  |
| 12     | Fitness Tracker       | MicroMax                             | Charge 4               | 214        | 13000  |
| 13     | Camera Bag            | Canva                                | ProTactic BP 350 AW II | 220        | 1200   |
| 14     | Smart Home Hub        | Google                               | Nest Hub               | 231        | 8000   |
| 15     | Power Bank            | SamSung                              | JP-101                 | 314        | 2500   |
| 16     | USB C                 | SAMSUNG                              | T-150                  | 56         | 550    |
| 17     | case                  | oppo                                 | 78                     | 88         | 7800   |



```
Serial No you want to change
1
```


```

```
1  
The 1 Star Customer Is: With Purchase count 9  
POI  
The 2 Star Customer Is: With Purchase count 4  
OIU
```

2
The Payment's done by method
Card - Debit Is: 7
The Payment's done by method
Cash Is: 17
The Payment's done by method
UPI Is: 11
The Payment's done by method
Card - Credit Is: 1

4
Amount_Given_By_All_Customers:
(1476147.0,)

10,10 2010
7
Thank You Customer For FeedBackiu
Our Rating By iu is 4 Out of 5
Thank You Customer For FeedBackyu
Our Rating By yu is 4 Out of 5
hi
io
Thank You Customer For FeedBack CNC
Our Rating By CNC is 5 Out of 5
super
Thank You Customer For FeedBack ij
Our Rating By ij is 5 Out of 5
ui0
Thank You Customer For FeedBack SANJEEV
Our Rating By SANJEEV is 5 Out of 5
test
Thank You Customer For FeedBack ufugefgue
Our Rating By ufugefgue is 4 Out of 5
badasss\
Thank You Customer For FeedBack kjdkjkdgh
Our Rating By kjdkjkdgh is 4 Out of 5
bfgdfgd
Thank You Customer For FeedBack SANJEEV
Our Rating By SANJEEV is 5 Out of 5

5

Year: 2023, Month: 1, Total Salary: 901890.0

Year: 2023, Month: 11, Total Salary: 574257.0

8

Serial Amount	Date	Customer Name	Phone Number	Payment Type	Item Name	Discount	Final
51 2400.0	2023-11-18	WDFGUFR	+912467843225	Card - Debit	Phone Case	4	

Welcome Admin ANSHUMAN MJ

What Do You Want To Modify

9

Date Format = 20xx-0x(month)-xx(date)

Enter Starting date You want To Search: 11-02-2023

Enter Ending date You want to Search: 12-12-2023

NO CUSTOMER BETWEEN THAT DATE RANGE

Or If You Want To Quit Press q

SUGGESTED IMPROVEMENTS

With additional time, we can elevate the codebase's quality by prioritizing readability through meaningful variable names and comments, fostering an environment of easier comprehension. A robust testing suite, encompassing diverse scenarios, will fortify the code's reliability. Time invested in performance optimization will address bottlenecks, enhancing system efficiency. Implementing comprehensive error handling mechanisms is crucial for robustness. Lastly, improving documentation will offer a clearer insight into the code's architecture, functions, and usage, streamlining onboarding for new team members and ensuring long-term maintainability.

BIBLIOGRAPHY

- Wikipedia
- Sumitha Arora Textbook for Class 11 Computer Science
- Sumitha Arora Textbook for Class 12 Computer Science
- w3schools