

ELECTRONICS SHOP *MANAGEMENT*



*MADE BY – VAIBHAV PATHAK
DINESH GROVER
OJAS CHAUDHARY*

CERTIFICATE

This is to certify that the project entitled “**ELECTRONICS SHOP MANAGEMENT**”, which deals making bills and customer accounts and product handling is submitted by **Dinesh Grover, Ojas Chaudhary** and **Vaibhav Pathak**. This project is bona fide piece of work carried out with the consultation of supervisor.

Vaibhav Pathak

Dinesh Grover

Ojas Chaudhary

XII

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would not be completed without mentioning the name of the people whose constant guidance and encouragement has crowned all our effort with success.

“**THANK YOU**”. These two words are very less to be measured when it comes to extend our gratitude towards all those who have made our project a memorable experience.

We own our sincere and whole hearted thanks to **MVN SCHOOL** who allowed us to make project report on **ELECTRONIC SHOP MANAGEMENT**. Also we are grateful to **MS.PRAGYA**, our teacher for constantly guiding us and tackling a variety of hurdle with implicit patience throughout our research project infused in us a great inspiration and confidence in taking up this study in right direction.

Words cannot express our sincere thanks to **CBSE BOARD** who had been a constant source of guidance and give us a chance to make a project.

Also we are sincerely thankful to all the respondents of the questionnaire who spared their valuable time. And last but not the least we would like to convey our heartiest thanks to our parents and friends with whose support and guidance this assignment of ours is successfully accomplished.

Vaibhav Pathak

Dinesh Grover

Ojas Chaudhary

XII

Contents

❖ System Description at Glance

❖ Database design and structure

❖ Python program :

- Front End code

- Back End code

System Description at a Glance

Electronic shop management provides a system that contains table for a product, customer, invoice, warranty, offers. This system provides access to customer and staff of the shop & the admin. It is a system that can make bills and customer account and auto generate bill amount. These are following activities in the management system:

- 1) Enter as an administrator, employee or a customer
- 2) Add and update records by login as an administrator or employee
- 3) Create an invoice and customer account for the customer and apply given offers
- 4) Show the customer invoice by mobile number verification when entered as a customer

First the user needs to log in as one of the before mentioned categories. Now the user can add or update data from any table when entered as an admin. When entered an employee the user can create a customer account and invoice and also hold the right to see all records in all the 5 tables. Customer can see their invoice and offers available to him when entered as customer. There exists a password protection when entering as an administrator or employee.

MYSQL

(Contains Five Interlinked Tables)

1)Table products:

Field	Type	Null	Key	Default	Extra
s_no	int	NO		NULL	
name	varchar(20)	NO		NULL	
model_no	varchar(20)	NO	PRI	NULL	
price	int	NO		NULL	

```
create table products(  
s_no int NOT NULL,  
name varchar(20) NOT NULL,  
model_no varchar(20) PRIMARY KEY,  
price int NOT NULL);
```

2)Table Invoice:

Field	Type	Null	Key	Default	Extra
invoice_no	int	NO	PRI	NULL	
product1	varchar(20)	NO	MUL	NULL	
product2	varchar(20)	YES	MUL	_cp850\ '00000000000000000000\ '	DEFAULT_GENERATED
product3	varchar(20)	YES	MUL	_cp850\ '00000000000000000000\ '	DEFAULT_GENERATED
bill_amt	int	NO		NULL	
DOP	date	YES		NULL	

```
create table invoice(  
invoice_no int PRIMARY KEY,  
product1 varchar(20) NOT NULL,  
product2 varchar(20) DEFAULT('00000000000000000000'),  
product3 varchar(20) DEFAULT('00000000000000000000'),  
bill_amt int NOT NULL,  
DOP date,  
FOREIGN KEY(product1) REFERENCES products(model_no),  
FOREIGN KEY(product2) REFERENCES products(model_no),  
FOREIGN KEY(product3) REFERENCES products(model_no));
```

3)Table Customer:

Field	Type	Null	Key	Default	Extra
customer_ID	varchar(6)	NO	PRI	NULL	
name	varchar(20)	NO		NULL	
mobile_num	varchar(10)	NO		NULL	
invoice_no	int	NO	MUL	NULL	

```
create table customer(  
customer_ID varchar(6) PRIMARY KEY,  
name varchar(20) NOT NULL,  
mobile_num varchar(10) NOT NULL,  
invoice_no int NOT NULL,  
FOREIGN KEY(invoice_no) REFERENCES invoice(invoice_no));
```

4)Table Warranty:

Field	Type	Null	Key	Default	Extra
product_name	varchar(20)	NO		NULL	
model_no	varchar(20)	NO	PRI	NULL	
start_date	date	YES		sysdate()	DEFAULT_GENERATED
end_date	date	YES		(`start_date` + interval 2 year)	DEFAULT_GENERATED

```
create table warranty(  
product_name varchar(20) NOT NULL,  
model_no varchar(20) PRIMARY KEY,  
start_date date DEFAULT(sysdate()),  
end_date date DEFAULT(DATE_ADD(start_date,INTERVAL 2 YEAR)),  
FOREIGN KEY(model_no) REFERENCES products(model_no),  
CONSTRAINT CHK_DATE CHECK(end_date>=DATE_ADD(start_date,INTERVAL 2 YEAR)));
```


5)Table Offers:

Field	Type	Null	Key	Default	Extra
off_code	varchar(4)	NO	PRI	NULL	DEFAULT_GENERATED
Discount_per	int	YES		0	
model_no	varchar(20)	NO	MUL	NULL	
p_name	varchar(20)	NO		NULL	

```
create table offers(  
off_code varchar(4) PRIMARY KEY,  
Discount_per int DEFAULT(0),  
model_no varchar(20) NOT NULL,  
p_name varchar(20) NOT NULL,  
FOREIGN KEY(model_no) REFERENCES products(model_no));
```

PYTHON

(Contains Two Modules)

Front End

```
#running module

import mysql.connector as myconn

import BACK_END as pack

mycon=myconn.connect(host='localhost',user='root',passwd='vvvvvvvp14',database='project')

print('_____')

print()

print('WELCOME TO GADA ELECTRONICS')

print('_____')

def admin(b):

    if b==1:

        print('in table Products')

        pack.add_p()

    elif b==2:

        print('in table Customer')

        pack.add_cust()

    elif b==3:

        print('in table Invoice')

        pack.add_invoice()

    elif b==4:

        print('in table Warranty')

        pack.add_warranty()

    elif b==5:

        print('in table offers')

        pack.add_offer()

    elif b==6:

        pack.update_tables()

    elif b==0:

        loop()
```

```
def employee(b):
```

```
    if b==1:
```

```
        print('in table Customer')
```

```
        pack.add_cust()
```

```
    elif b==2:
```

```
        print('in table Invoice')
```

```
        pack.add_invoice()
```

```
    elif b==3:
```

```
        print('Products')
```

```
        pack.show_p()
```

```
    elif b==4:
```

```
        print('Customers')
```

```
        pack.show_cust()
```

```
    elif b==5:
```

```
        print('Invoices')
```

```
        pack.show_invoice()
```

```
    elif b==6:
```

```
        print('Warranty')
```

```
        pack.show_warranty()
```

```
    elif b==7:
```

```
        print('Offers')
```

```
        pack.show_offer()
```

```
    elif b==0:
```

```
        loop()
```

```
def cust(b):
```

```
    if b==1:
```

```
        pack.show_cust_invoice()
```

```
    elif b==2:
```

```
        pack.show_offer()
```

```
    elif b==0:
```

```
        loop()
```

#show_invoice can only be operated by employee

#show_cust_invoice shows only one user record

def loop():

flag=False

a=int(input('Enter as

1.Admin

2.Employee

3.Customer

\nenter 0 to close\t:')

while a!=0:

if a==1:

if flag==False:

passwd=int(input('enter Admin password'))

if passwd==706968:

print(''

Welcome back Admin')

flag=True

else:

print('WRONG PASSWORD')

continue

b=int(input(''

enter 1 to add product

enter 2 to add customer

enter 3 to add invoice

enter 4 to add warranty

enter 5 to add offer

enter 6 to update tables

enter 0 to return\t')

admin(b)

#loop(a)

elif a==2:

if flag==False:

```
passwd=int(input('enter employee password'))
```

```
if passwd==345600:
```

```
    print("""
```

```
Welcome back employee""')
```

```
    flag=True
```

```
else:
```

```
    print('WRONG PASSWORD')
```

```
    continue
```

```
    #using loop in employee
```

```
    b=int(input("""
```

```
enter 1 to add customer
```

```
enter 2 to add invoice
```

```
enter 3 to show products
```

```
enter 4 to show customer
```

```
enter 5 to show invoice
```

```
enter 6 to show warranty
```

```
enter 7 to show offers
```

```
enter 0 to return\t:""))
```

```
        employee(b)
```

```
    elif a==3:
```

```
        b=int(input("""
```

```
enter 1 to see your invoice
```

```
enter 2 to see all offers
```

```
or enter 0 to return""'))
```

```
        cust(b)
```

```
loop()
```

Back End

```
import mysql.connector as myconn

mycon=myconn.connect(host='localhost',user='root',passwd='vvvvvvvp14',database='project')

def auto_s_no():

    #to auto generate s number when used

    cursor=mycon.cursor()

    cursor.execute('select MAX(s_no) from products')

    data=cursor.fetchall()

    s_no=0

    for a in data:

        s_no=a[0]+1

    return s_no

def auto_cust_ID():

    #to auto fill the customer ID column when used

    cursor=mycon.cursor()

    cursor.execute('select MAX(customer_ID) from customer')

    data=cursor.fetchall()

    for a in data:

        cust_ID=""

        for b in range(2,len(a[0])):

            cust_ID+=a[0][b]

        cust_ID=int(cust_ID)+1

        cust_ID=str(cust_ID)

        if len(cust_ID)==1:

            cust_ID='GE000'+cust_ID

        elif len(cust_ID)==2:

            cust_ID='GE00'+cust_ID

        elif len(cust_ID)==3:

            cust_ID='GE0'+cust_ID

        cust_ID='GE'+cust_ID
```

```
elif len(cust_ID)==4:
```

```
    cust_ID='GE'+cust_ID
```

```
else:
```

```
    cust_ID='ERROR(customer ID too long)'
```

```
return cust_ID
```

```
def auto_invoice_no():
```

```
#to auto generate invoice number when used
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute('select MAX(invoice_no) from invoice')
```

```
    data=cursor.fetchall()
```

```
    for a in data:
```

```
        return (a[0]+1)
```

```
def auto_bill_amt(a,b,c):
```

```
#to automatically fill the bill amount when used
```

```
    amt=0
```

```
    for q in [a,b,c]:
```

```
        cursor=mycon.cursor()
```

```
        price=('select price from products where model_no IN (%s)')
```

```
        data_price=(q,)
```

```
        cursor.execute(price,data_price)
```

```
        data=cursor.fetchall()
```

```
        amt+=data[0][0]
```

```
    return amt
```

```
def update_tables():
```

```
    cursor=mycon.cursor()
```

```
    print("\n*****Update*****")
```

```
    Currently entered data in tables of database")
```

```
    table=int(input("enter 1 for products
```

```
enter 2 for customer
```

```
enter 3 for invoice
```

```
enter 4 for offers \t:"))
```


if table==1:

```
print("\n\nIn PRODUCTS you can only update the following column(s)
```

```
*****price*****")
```

```
a=int(input("\nDo you want to a specific row or whole column(s) to be updated
```

```
enter 1 for row
```

```
enter 0 for whole column(s)\t :")
```

```
if a==1:
```

```
    update_products=('UPDATE PRODUCTS SET price=%s WHERE model_no=%s')
```

```
    model_no=input('\nenter the model no. of the product\t:')
```

```
    new_price=int(input('enter new price\t\t :'))
```

```
    data_products=(new_price,model_no)
```

```
    cursor.execute(update_products,data_products)
```

```
    mycon.commit()
```

```
elif a!=1:
```

```
    update_products=("UPDATE PRODUCTS SET price=%s * price")
```

```
    new_price_percent=input('\nenter new price increase inpercent:')
```

```
    new_price_percent=int(new_price_percent)
```

```
    new_price_percent=(new_price_percent+100)/100
```

```
    new_price_percent=str(new_price_percent)
```

```
    print(new_price_percent,type(new_price_percent))
```

```
    update_products=('update PRODUCTS SET price='+new_price_percent+'*price')
```

```
    cursor.execute(update_products,new_price_percent)
```

```
    mycon.commit()
```

```
elif table==2:
```

```
print("\n\nIn CUSTOMER you can only update the following column(s)
```

```
*****name,mobile number*****
```

```
And the update can only be made to a row at once")
```

```
a=int(input("\nDo you want to update both or one of those column(s)
```

```
enter 1 for name only
```

```
enter 2 for mobile number only
```

enter 0 for both\t:''))

if a==1:

update_customer=('UPDATE CUSTOMER SET name=%s WHERE customer_ID=%s')

customer_ID=input('\nenter the customer_ID\t:')

new_name=input('enter new name\t:')

data_customer=(new_name,customer_ID)

cursor.execute(update_customer,data_customer)

mycon.commit()

elif a==2:

update_product=('UPDATE CUSTOMER SET mobile_num=%s WHERE customer_ID=%s')

customer_ID=input('\nenter the customer_ID\t:')

new_mobile_num=int(input('enter new mobile no\t:'))

data_customer=(new_mobile_num,customer_ID)

cursor.execute(update_customer,data_customer)

mycon.commit()

elif a==0:

update_product=('UPDATE CUSTOMER SET name=%s,mobile_num=%s WHERE
customer_ID=%s')

customer_ID=input('\nenter the customer_ID\t:')

new_name=input('enter new name\t:')

new_mobile_num=int(input('enter new mobile no\t:'))

data_customer=(new_name,new_mobile_num,customer_ID)

cursor.execute(update_customer,data_customer)

mycon.commit()

elif table==3:

print("\n\n INVOICE you can only update the following column(s)

*****DOP*****

And the update can only be made to a row at once")

update_invoice=('UPDATE INVOICE SET DOP=%s WHERE invoice_no=%s')

invoice_no=input('\nenter the invoice_no\t:')

new_DOP=input('enter new DOP\t:')

data_invoice=(new_DOP,invoice_no)

```
cursor.execute(update_invoice,data_invoice)
```

```
mycon.commit()
```

```
elif table==4:
```

```
print("\n\n OFFERS you can only update the following column(s)
```

```
*****Discount_per*****
```

```
And the update can be to a row only at once")
```

```
update_offers=('UPDATE offers SET Discount_per=%s WHERE off_code=%s')
```

```
off_code=input('\nenter off_code\t:')
```

```
new_Discount_per=int(input('enter the Discount_per\t:'))
```

```
data_offers=(new_Discount_per,off_code)
```

```
cursor.execute(update_offers,data_offers)
```

```
mycon.commit()
```

```
else:
```

```
print('\n*****NOT*A*TABLE*YOU*CAN*UPDATE*****')
```

```
def add_p():
```

```
    #s_no to be auto generated
```

```
cursor=mycon.cursor()
```

```
add_product=('INSERT INTO PRODUCTS'
```

```
    '(s_no, name,model_no,price)'
```

```
    'values(%s,%s,%s,%s)')
```

```
s_no=auto_s_no()
```

```
print('Serial number\t:',s_no)
```

```
name=input('enter name of product')
```

```
model_no=input('enter model number of product')
```

```
price=int(input('enter price of product'))
```

```
data_product=(s_no,name ,model_no,price)
```

```
#insertion into table
```

```
cursor.execute(add_product,data_product)
```

```
mycon.commit()
```

```
print('executed')
```

```
#insertion of a customer
```

```
invoice_no=0
```

```
def add_cust():
```

```
    global invoice_no
```

```
    add_customer=('INSERT INTO CUSTOMER'
```

```
                '(invoice_no,name,mobile_num,customer_id)'
```

```
                'values(%s,%s,%s,%s)')
```

```
    invoice_no=int(input('enter invoice number :'))
```

```
    name=input('enter customer name :')
```

```
    mobile_num=input('enter mobile number :')
```

```
    cust_id=auto_cust_ID()
```

```
    print('Customer ID\t\t:',cust_id)
```

```
    data_customer=(invoice_no,name,mobile_num,cust_id)
```

```
    #insertion into table
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute(add_customer,data_customer)
```

```
    mycon.commit()
```

```
    print("executed")
```

```
def add_invoice():
```

```
    #global invoice_no
```

```
    #bill amount and invoice number auto generated
```

```
    num=int(input('enter number of product(1-3)'))
```

```
    invoice_no=auto_invoice_no()
```

```
    print('Invoice number\t:',invoice_no)
```

```
    product1=input('enter model number of product 1:')
```

```
    #arranging invoice according to product purchased
```

```
    if num==3:
```

```
        add_in=('INSERT INTO INVOICE'
```

```
                '(invoice_no,product1,product2,product3,bill_amt,dop)'
```

```
                ' values(%s,%s,%s,%s,%s,%s)')
```

```
product2=input('enter model number of product 2:')
```

```
product3=input('enter model number of product 3:')
```

```
bill_amt=auto_bill_amt(product1,product2,product3)
```

```
dop=input('enter date of purchase      :')
```

```
data_in=(invoice_no,product1,product2,product3,bill_amt,dop)
```

```
print('Amount to be paid\t: ',bill_amt)
```

```
elif num==2:
```

```
add_in=('INSERT INTO INVOICE'
```

```
      '(invoice_no,product1,product2,bill_amt,dop)'
```

```
      ' values(%s,%s,%s,%s,%s)')
```

```
product2=input('enter model number of product 2:')
```

```
bill_amt=auto_bill_amt(product1,product2,'00000000000000000000')
```

```
dop=input('enter date of purchase      :')
```

```
data_in=(invoice_no,product1,product2,bill_amt,dop)
```

```
print('Amount to be paid\t: ',bill_amt)
```

```
elif num==1:
```

```
add_in=('INSERT INTO INVOICE'
```

```
      '(invoice_no,product1,bill_amt,dop)'
```

```
      ' values(%s,%s,%s,%s)')
```

```
bill_amt=auto_bill_amt(product1,'00000000000000000000','00000000000000000000')
```

```
dop=input('enter date of purchase      :')
```

```
data_in=(invoice_no,product1,bill_amt,dop)
```

```
print('Amount to be paid\t: ',bill_amt)
```

```
else:
```

```
    print('enter number between 1-3')
```

```
    add_invoice()
```

```
cursor=mycon.cursor()
```

```
cursor.execute(add_in,data_in)
```

```
mycon.commit()
```

```
print("executed")
```

```
def add_warranty():
```

```
cursor=mycon.cursor()
```

```
year=int(input('"enter number of years of warranty  
or press enter 0 for current date"))
```

```
p_name=input('enter name of product')
```

```
model_no=input('enter model number of product')
```

```
if year==0:
```

```
    add_warranty=('INSERT INTO warranty'
```

```
                '(product_name,model_no)'
```

```
                'values(%s,%s)')
```

```
    data_warranty=(p_name,model_no)
```

```
elif year!=0:
```

```
    end_date=input('enter the end date')
```

```
    add_warranty=('INSERT INTO warranty'
```

```
                '(product_name,model_no,end_date)'
```

```
                'values(%s,%s,%s)')
```

```
    data_warranty=(p_name,model_no,end_date)
```

```
cursor.execute(add_warranty,data_warranty)
```

```
mycon.commit()
```

```
print("executed")
```

```
def add_offer():
```

```
    add_offer=('INSERT INTO offers VALUES(%s,%s,%s,%s)')
```

```
    off_code=input('enter offer code  :')
```

```
    off_name=input('enter offer name  :')
```

```
    model_no=input('enter model number  :')
```

```
    Discount_per=int(input('enter Discount percentage:'))
```

```
    data_offers=(off_code,Discount_per,model_no,off_name)
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute(add_offer,data_offers)
```

```
    mycon.commit()
```

```
    print("executed")
```

```
def show_p():
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute('select * from products order by s_no')
```

```
    data=cursor.fetchall()
```

```
    for row in data:
```

```
        print(row)
```

```
def show_cust():
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute('select * from customer order by customer_id')
```

```
    data=cursor.fetchall()
```

```
    for row in data:
```

```
        print(row)
```

```
def show_invoice():
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute('select * from invoice')
```

```
    data=cursor.fetchall()
```

```
    for row in data:
```

```
        print(row)
```

```
def show_offer():
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute('select * from offers order by off_code')
```

```
    data=cursor.fetchall()
```

```
    for row in data:
```

```
        print(row)
```

```
def show_warranty():
```

```
    cursor=mycon.cursor()
```

```
    cursor.execute('select * from warranty')
```

```
    data=cursor.fetchall()
```

```
    for row in data:
```

```
        print(row)
```

```
def show_cust_invoice():
```

```
mob_no=input('Confirm your mobile_no')
```

```
cursor=mycon.cursor()
```

```
cursor.execute("select * from customer where mobile_num="+mob_no)
```

```
fake_data=cursor.fetchone()
```

```
invoice_no_ex=fake_data[0]
```

```
#showing invoice from invoice number extracted
```

```
invoice_no_ex=str(invoice_no_ex)
```

```
cursor=mycon.cursor()
```

```
cursor.execute("select * from invoice where invoice_no="+invoice_no_ex)
```

```
data=cursor.fetchone()
```

```
print(data)
```
