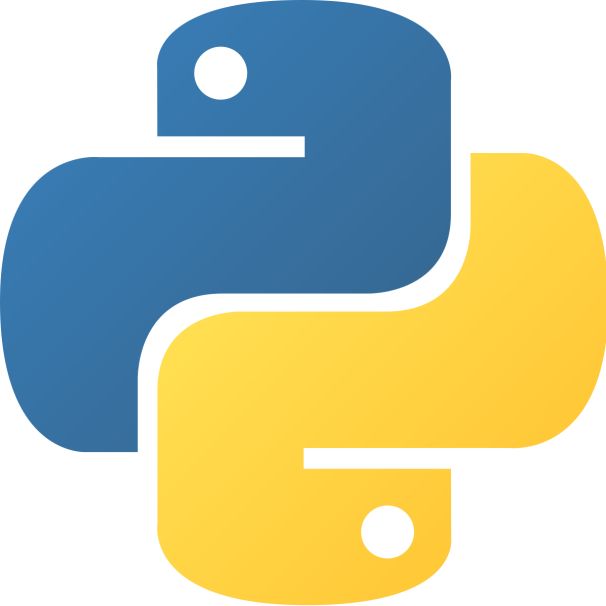
***ELECTRONICS SHOP***

***MANAGEMENT***





*MADE BY –DINESH GROVER*

*OJAS CHAUDHARY*

*VAIBHAV PATHAK*

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.

**Dinesh Grover**

**Ojas Chaudhary**

**Vaibhav Pathak**

**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 crowed 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.

**Dinesh Grover**

**Ojas Chaudhary**

**Vaibhav Pathak**

**XII**

*D*

***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

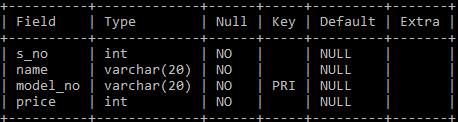
4) Show the customer invoice by mobile number verification when entered as a customer

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:



create table products(

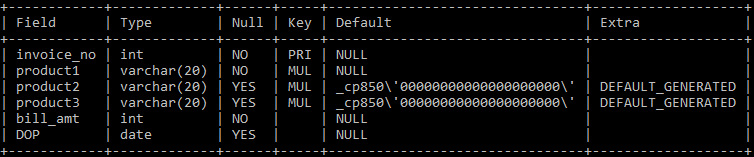
s\_noint NOT NULL,

namevarchar(20) NOT NULL,

model\_novarchar(20) PRIMARY KEY,

priceint NOT NULL);

2)Table Invoice:



create table invoice(

invoice\_noint PRIMARY KEY,

product1varchar(20) NOT NULL,

product2varchar(20) DEFAULT('00000000000000000000'),

product3varchar(20) DEFAULT('00000000000000000000'),

bill\_amtint NOT NULL,

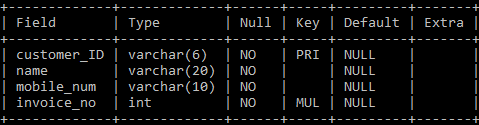
DOP date,

FOREIGN KEY(product1) REFERENCES products(model\_no),

FOREIGN KEY(product2) REFERENCES products(model\_no),

FOREIGN KEY(product3) REFERENCES products(model\_no));

3)Table Customer:



create table customer(

customer\_IDvarchar(6) PRIMARY KEY,

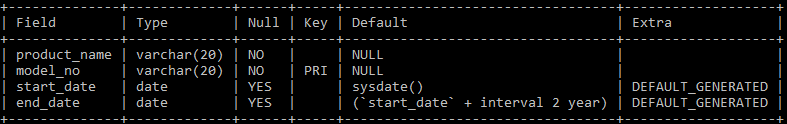
namevarchar(20) NOT NULL,

mobile\_numvarchar(10) NOT NULL,

invoice\_noint NOT NULL,

FOREIGN KEY(invoice\_no) REFERENCES invoice(invoice\_no));

4)Table Warranty:



create table warranty(

product\_namevarchar(20) NOT NULL,

model\_novarchar(20) PRIMARY KEY,

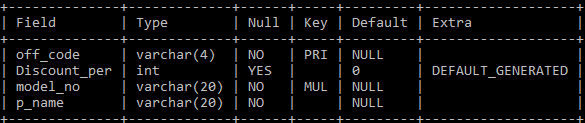
start\_date date DEFAULT(sysdate()),

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:



create table offers(

off\_codevarchar(4) PRIMARY KEY,

Discount\_perintDEFAULT(0),

model\_novarchar(20) NOT NULL,

p\_namevarchar(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

#using loop

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

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('''\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 :'))

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('''\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('''\nIn 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('''\nIn 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 :',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()

#insertion into table

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():

#in progress

#

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)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_