

“SALOON MANAGEMENT SYSTEM”

**A COMPUTER SCIENCE
PROJECT REPORT
SUBMITTED BY
AMEE DADHANIYA
IN FULFILMENT OF THE
PROJECT WORK ASSIGNED
BY AISSCE – (2022-2023)
IN
COMPUTER SCIENCE (083)
AT**



J.B. DIAMOND & KARP IMPEX VIDYA SANKUL



J.B. Diamonds & KARP Impex Vidya Sankul

Opp. Diamond Nagar, B/H Thakor Dwar Farm, Surat - Kamrej Road, Laskana

Phone No: 9228025712, Email id: jbkarpschool.cbse@gmail.com

Web: www.jbkarpschool.ac.in

CBSE-English Medium



This is to certify that **Ms AMEE DADHANIYA** is a student of

J. B. DIAMOND & KARP IMPEX Vidya Sankul, who has successfully completed the project work on title **“SALOON MANAGEMENT SYSTEM”** in **COMPUTER SCIENCE (083)** assigned to her as a part of AISSCE curriculum during the academic year **2022-23**.

We found her very sincere, hardworking and disciplined girl.

We wish all the success for her future endeavors.

Signature of the Internal Examiner

Signature of the External Examiner

**Principal
Signature**



PROJECT FILE



ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my Computer Science teacher **Mr. Ajay Tiwari sir** as well as our principal **Mr. Gaurang Patel Sir** for their guidance and support in completing this wonderful project entitled **“SALOON MANAGEMENT SYSTEM”** using **Python-MySQL connectivity**”.

I came to know about many new things.
I am really thankful to them.

A debt of gratitude is also owed to my parents and friends who helped me with their valuable suggestions.

Although this report has been prepared with utmost care and deep routed interest, even then I accept respondents and imperfections.

Contents

S.No.	Topics
1.	Aim
2.	Introduction
3.	Python Coding
4.	Database Structure
5.	Input-Output Interference
6.	Bibliography



Aim

**SALOON'DATA
MANAGEMENT**
Using MYSQL connectivity



Introduction

- **What is Python?**

- The Python Programming Language is a recent, general-purpose and higher-level programming language. It has features for database programming also.
- This project aims on explaining how one can create a MySQL database from within a Python script and create a user interface software.

- **Why Python?**

- Due to its open source nature, Python has been ported to many platforms.
- It is free and open source. It is available for free and runs on almost every current platform.
- Python provides interfaces to all major commercial databases.
- It can easily integrated with C, C++, COM, Java, MySQL, etc.

- **What is MySQL?**

- MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).
- It provides you with a rich set of features that support a secure environment for storing, maintaining, and accessing data.

- **Why MySQL?**

- It is an open source software and is easily portable.
- It is easy to use, manage and works quickly and efficiently.
- It is used to create databases, manage security of a database.
- It maintains integrity and reduces data redundancy.



Python is a
Front End
Software

MySQL is a
Back End
Software

Interface Python with MySQL

There are mainly seven steps that must be followed in order to create a database connectivity application.

Step 1 – Start Python

Step 2 – Import the packages required for database programming.

Step 3 – Open a connection to database.

Step 4 – Create a cursor instance.

Step 5 – Execute a query.

Step 6 – Extract data from result set.

Step 7 – Clean up the environment.



Python Coding

SOURCE CODE

LOGIN FORM

#LOGIN FORM

```
from tkinter import*
import tkinter.messagebox          # for messagebox
import os                          # for stringvariable
from tkinter import ttk           # for combobox
import random                      # for reference
import time
import datetime

def main():
    root = Tk()
    app = Window_1(root)

class Window_1:
    def __init__(self, master):
        self.master = master
        self.master.title("Login Window")
        self.master.geometry('750x550')
        self.master.config(bg = 'tan')
        self.Frame = Frame(self.master, bg = 'tan')
        self.Frame.pack()

        self.Username = StringVar()          # x = StringVar() Holds a string; default value is "
        self.Password = StringVar()

        self.Lbl_Title = Label(self.Frame, text = 'Login Form', font = ('algerian',55,'underline'), bg =
'tan', fg = 'red')
        self.Lbl_Title.grid(row = 0, column = 0, columnspan =3, pady = 40)

        self.Login_Frame_1 = LabelFrame(self.Frame, width = 1350, height = 150, relief = 'ridge', bg
```

```
= 'mistyrose', bd = 13, text='Login', fg = 'darkblue',
        font = ('lucida calligraphy',25,'bold'))
self.Login_Frame_1.grid(row = 1, column =0)
self.Login_Frame_2 = LabelFrame(self.Frame, width = 1000, height = 150, relief = 'ridge',bg =
'powderblue', bd = 15, text='Events', fg = 'darkblue',
        font = ('lucida calligraphy',25,'bold'))
self.Login_Frame_2.grid(row = 2, column = 0)
```

```
#=====LABEL and
ENTRIES=====
self.Label_Username = Label(self.Login_Frame_1, text = 'Username', font = ('lucida
handwriting',20,'bold'), bg = 'mistyrose', fg = 'black', bd = 20)
self.Label_Username.grid(row = 0, column = 0)
self.text_Username = Entry(self.Login_Frame_1, font = ('lucida handwriting',20,'bold'),
fg="red",textvariable = self.Username)
self.text_Username.grid(row = 0, column = 1, padx = 50)
self.text_Username.focus()

self.Label_Password = Label(self.Login_Frame_1, text = 'Password', font = ('lucida
handwriting',20,'bold'), bg = 'mistyrose', fg = 'black', bd = 20)
self.Label_Password.grid(row = 1, column = 0)
self.text_Password = Entry(self.Login_Frame_1, font = ('lucida handwriting',20,'bold'), show
= '~', fg="green", textvariable = self.Password)
self.text_Password.grid(row = 1, column = 1)
```

```
#=====BUTTONS=====
=====
self.btnLogin = Button(self.Login_Frame_2, text = 'Login', fg = 'green', width = 10, font =
('lucida calligraphy',19,'bold'), command = self.Login)
self.btnLogin.grid(row = 3, column = 0, padx = 8, pady = 20)

self.btnReset = Button(self.Login_Frame_2, text = 'Reset', fg = 'black', width = 10, font =
('lucida calligraphy',19,'bold'), command = self.Reset)
self.btnReset.grid(row = 3, column = 1, padx = 8, pady = 20)

self.btnExit = Button(self.Login_Frame_2, text = 'Exit', fg = 'red', width = 10, font = ('lucida
calligraphy',19,'bold'), command = self.Exit)
```

```
self.btnExit.grid(row = 3, column = 2, padx = 8, pady = 20)
```

```
#=====Code for Login
```

```
Button=====
```

```
def Login(self):
```

```
    u = (self.Username.get())
```

```
    p = (self.Password.get())
```

```
    if (u == str('root') and p == str(12345)):
```

```
        tkinter.messagebox.askyesno("Login Successfully", "Thanks : For using Login Form.")
```

```
        self.master.destroy()
```

```
        self.__library__()
```

```
    else:
```

```
        tkinter.messagebox.askyesno("Login", "Error : Wrong Password")
```

```
        self.Username.set("")
```

```
        self.Password.set("")
```

```
        self.text_Username.focus()
```

```
#=====Code for Reset
```

```
Button=====
```

```
def Reset(self):
```

```
    self.Username.set("")
```

```
    self.Password.set("")
```

```
    self.text_Username.focus()
```

```
#=====Code for Exit
```

```
Button=====
```

```
def Exit(self):
```

```
    self.Exit = tkinter.messagebox.askokcancel("Login System", "Confirm if you want to Exit")
```

```
    if self.Exit > 0:
```

```
        self.master.destroy()
```

```
def __library__(self):
```

```
    filename = 'blood.py'
```

```
    os.system(filename)
```

```
os.system('notepad'+filename)
```

```
if __name__ == '__main__':  
self-and-__init__-method-in-python-class/  
    main()
```

<https://micropyramid.com/blog/understand->

SOURCE CODE

```
from tkinter import *  
from tkinter import ttk  
from tkinter import messagebox  
import time  
import mysql.connector as sql  
from subprocess import call  
import datetime as dt  
import tkinter  
  
def SplashScreen():  
    splashscreen = Tk()  
    splashscreen.overridedirect(1)  
    # Remove Title Bar  
    splashscreen.geometry(  
        f"800x400+{(splashscreen.winfo_screenwidth() - 800) //  
2}+{(splashscreen.winfo_screenheight() - 400) // 2}")  
    splashscreen.configure(bg='powderblue')  
    Label(splashscreen, text='WELCOME TO ELEGANCE', font='Impact 40', fg='orange',  
bg='powderblue').pack()  
    Label(splashscreen, text="The Saloon", font=('Bold', 22), bg='powderblue',  
fg='orange').place(x=570, y=60)  
    pgbar = ttk.Progressbar(splashscreen, orient='horizontal', length=600, mode='determinate')  
    Label(splashscreen, text="Designed by : Amee Dadhaniya", font=('algerian', 15),  
bg='powderblue', fg='brown').place(x=390, y=300)  
    Label(splashscreen, text="12 Commerce ", font=('algerian',15), bg='powderblue',  
fg='brown').place(x=590, y=330)  
    pgbar.place(x=100, y=150)  
    pgbar['maximum'] = 100
```

```
txt=Label(splashscreen,text='0%',relief=GROOVE,bg='orange',fg='orange',font=('lucida'))
txt.place(x=705, y=151)
```

```
for i in range(101):
```

```
    time.sleep(0.001)
    pgbar['value'] = i
    pgbar.update()
    txt['text']=pgbar['value'],'%'
    splashscreen.destroy()
    splashscreen.mainloop()
```

```
mydb=mysql.connect(host="localhost",user="root",password="12345")#connection to mysql
mycur=mydb.cursor()
mycur.execute("create database if not exists saloon")
mycur.execute("use saloon")
mycur.execute('create table if not exists customerdata(customer_id integer primary key, name
varchar(30), services char(3) , doa date, gender varchar(10), contact varchar(12))')
```

```
class customer:
```

```
    def __init__(self,root):
```

```
        self.root=root
        self.root.title("ELEGANCE FAMILY SALOON")
        self.root.geometry("1350x700+0+0")
        title=Label(self.root,text="ELEGANCE CUSTOMER'S
DATA",font=("Algerian",42,"bold"),bg="grey",fg="pink")
        title.pack(side=TOP,fill=X)
```

```
##### All variable#####
```

```
    self.customer_id_var=StringVar()
    self.name_var=StringVar()
    self.services_var=StringVar()
    self.doa_var=StringVar()
    self.gender_var=StringVar()
    self.contact_var=StringVar()
    self.search_by=StringVar()
    self.search_txt=StringVar()
```

```
#####Manage Frame#####
```

```

Manage_frame=Frame(self.root,bd=4,relief=RIDGE,bg="orange")
Manage_frame.place(x=20,y=80,width=500,height=620)
m_title=Label(Manage_frame,text="Manage
Records",bg="orange",fg="white",font=("Maiandra GD",30,"bold underline"))
m_title.grid(row=0,columnspan=2,pady=15)

lbl_customer_id=Label(Manage_frame,text="CUSTOMER
ID",bg="pink",fg="black",font=("Maiandra GD",20,"bold"))
lbl_customer_id.grid(row=1,column=0,pady=10,padx=20,sticky="w")
txt_customer_id=Entry(Manage_frame,textvariable=self.customer_id_var,font=("Maiandra
GD",10,"bold"),bd=5,relief=GROOVE)
txt_customer_id.grid(row=1,column=1,pady=10,padx=20,sticky="w")

lbl_name=Label(Manage_frame,text="NAME",bg="pink",fg="black",font=("Maiandra
GD",20,"bold"))
lbl_name.grid(row=2,column=0,pady=10,padx=20,sticky="w")

txt_name=Entry(Manage_frame,textvariable=self.name_var,font=("Maiandra
GD",10,"bold"),bd=5,relief=GROOVE)
txt_name.grid(row=2,column=1,pady=10,padx=20,sticky="w")

lbl_services=Label(Manage_frame,text="SERVICES",bg="pink",fg="black",font=("Maiandra
GD",20,"bold"))
lbl_services.grid(row=3,column=0,pady=10,padx=20,sticky="w")

combo_services=ttk.Combobox(Manage_frame,textvariable=self.services_var,font=("Maiandra
GD",9,"bold"),state="readonly")
combo_services['values']=("spa","cut")
combo_services.grid(row=3,column=1,pady=10,padx=20,sticky="w")

lbl_doa=Label(Manage_frame,text="D.O.A",bg="pink",fg="black",font=("Maiandra
GD",20,"bold"))
lbl_doa.grid(row=4,column=0,pady=10,padx=20,sticky="w")

txt_doa=Entry(Manage_frame,textvariable=self.doa_var,font=("Maiandra
GD",10,"bold"),bd=5,relief=GROOVE)
txt_doa.grid(row=4,column=1,pady=10,padx=20,sticky="w")

```

```

    lbl_gender=Label(Manage_frame,text="GENDER",bg="pink",fg="black",font=("Maiandra
GD",20,"bold"))
    lbl_gender.grid(row=5,column=0,pady=10,padx=20,sticky="w")

    combo_gender=ttk.Combobox(Manage_frame,textvariable=self.gender_var,font=("Maiandra
GD",9,"bold"),state="readonly")
    combo_gender['values']=("Male","Female","Other")
    combo_gender.grid(row=5,column=1,pady=10,padx=20,sticky="w")

    lbl_contact=Label(Manage_frame,text="CONTACT",bg="pink",fg="black",font=("Maiandra
GD",20,"bold"))
    lbl_contact.grid(row=6,column=0,pady=10,padx=20,sticky="w")

    txt_contact=Entry(Manage_frame,textvariable=self.contact_var,font=("Maiandra
GD",10,"bold"),bd=5,relief=GROOVE)
    txt_contact.grid(row=6,column=1,pady=10,padx=20,sticky="w")


    #####button Frame#####
    btn_frame=Frame(Manage_frame,bd=4,relief=RIDGE,bg="powderblue")
    btn_frame.place(x=12,y=550,width=450)

    addbutton=Button(btn_frame,text="ADD",width=11,bg="white",fg="red",command=self.add_stu
dent).grid(row=0,column=0,padx=10,pady=10)

    updatebutton=Button(btn_frame,text="UPDATE",width=11,bg="white",fg="blue",command=self.
update_data).grid(row=0,column=1,padx=10,pady=10)

    deletebutton=Button(btn_frame,text="DELETE",width=12,bg="white",fg="green",command=self.
delete_data).grid(row=0,column=2,padx=10,pady=10)

    clearbutton=Button(btn_frame,text="CLEAR",width=12,bg="white",fg="purple",command=self.cl
ear).grid(row=0,column=3,padx=10,pady=10)


    #####Detail Frame#####
    Detail_frame=Frame(self.root,bd=4,relief=RIDGE,bg="brown")
    Detail_frame.place(x=530,y=80,width=790,height=580)

```

```

    lbl_search=Label(Detail_frame,text="Search By",bg="brown",fg="black",font=("Maiandra
GD",15,"bold"))
    lbl_search.grid(row=0,column=0,pady=10,padx=20,sticky="w")

combo_search=ttk.Combobox(Detail_frame,width=10,textvariable=self.search_by,font=("Maian
dra GD",10,"bold"),state="readonly")
    combo_search['values']=("customer_id","name","contact","services")

    combo_search.grid(row=0,column=1,pady=10,padx=20,sticky="w")
    txt_search=Entry(Detail_frame,textvariable=self.search_txt,font=("Maiandra
GD",10,"bold"),bd=4,relief=GROOVE)
    txt_search.grid(row=0,column=2,pady=10,padx=20,sticky="w")

searchbtn=Button(Detail_frame,text="Search",width=10,pady=3,command=self.search_data).gri
d(row=0,column=3,padx=10,pady=10)
    showbtn=Button(Detail_frame,text="Show
All",width=10,pady=3,command=self.fetch_data).grid(row=0,column=4,padx=10,pady=10)

#####Table Frame#####
Table_frame=Frame(Detail_frame,bd=4,relief=RIDGE,bg="orange2")
Table_frame.place(x=10,y=70,width=760,height=500)
scroll_x=Scrollbar(Table_frame,orient=HORIZONTAL)
scroll_y=Scrollbar(Table_frame,orient=VERTICAL)

self.customer_table=ttk.Treeview(Table_frame,columns=("customer_id","name","services","doa
","gender","contact"),xscrollcommand=scroll_x.set,yscrollcommand=scroll_y.set)
    scroll_x.pack(side=BOTTOM,fill=X)
    scroll_y.pack(side=RIGHT,fill=Y)
    scroll_x.config(command=self.customer_table.xview)
    scroll_y.config(command=self.customer_table.yview)

self.customer_table.heading("customer_id",text="Customer Id")
self.customer_table.heading("name",text="Name")
self.customer_table.heading("services",text="Services")
self.customer_table.heading("doa",text="D.O.A")
self.customer_table.heading("gender",text="Gender")
self.customer_table.heading("contact",text="Contact")

```

```

self.customer_table['show']='headings'
self.customer_table.column("customer_id",width=55)
self.customer_table.column("name",width=70)
self.customer_table.column("services",width=110)
self.customer_table.column("doa",width=70)
self.customer_table.column("gender",width=110)
self.customer_table.column("contact",width=110)
self.customer_table.pack(fill=BOTH,expand=1)
self.customer_table.bind("<ButtonRelease-1>",self.get_cursor)
self.fetch_data()
def add_student(self):
    if self.customer_id_var.get()==" or self.name_var.get()=="":
        messagebox.showerror("Error","All Fields are required!")
    else:
        mydb=sql.connect(host="localhost",user="root",password="12345",
database="saloon")#connection to mysql
        mycur=mydb.cursor()
        mycur.execute("insert into customerdata values(%s,%s,%s,%s,%s,%s);",(
            self.customer_id_var.get(),
            self.name_var.get(),
            self.services_var.get(),
            self.doa_var.get(),
            self.gender_var.get(),
            self.contact_var.get(),
        ))
        mydb.commit()
        self.fetch_data()
        messagebox.showinfo("Success","Record has been inserted")
        mydb.close()
def fetch_data(self):
    mydb=sql.connect(host="localhost",user="root",password="12345",
database="saloon")#connection to mysql
    mycur=mydb.cursor()
    mycur.execute("select * from customerdata")
    rows=mycur.fetchall()

```

```

if len(rows)!=0:
    self.customer_table.delete(*self.customer_table.get_children())
    for row in rows:
        self.customer_table.insert("",END,values=row)

mydb.commit()
mydb.close()
def clear(self):
    self.customer_id_var.set("")
    self.name_var.set("")
    self.services_var.set("")
    self.doa_var.set("")
    self.contact_var.set("")
    self.gender_var.set("")

def get_cursor(self,ev):
    cursr_row=self.customer_table.focus()
    contents=self.customer_table.item(cursr_row)
    row=contents['values']
    self.customer_id_var.set(row[0])
    self.name_var.set(row[1])
    self.services_var.set(row[2])
    self.doa_var.set(row[3])
    self.contact_var.set(row[5])
    self.gender_var.set(row[4])

def update_data(self):
    mydb=sql.connect(host="localhost",user="root",password="12345",
database="saloon")#connection to mysql
    mycur=mydb.cursor()
    mycur.execute("update customerdata set
name=%s,services=%s,doa=%s,gender=%s,contact=%s where customer_id=%s ;",(
        self.name_var.get(),
        self.services_var.get(),

```

```

        self.doa_var.get(),
        self.gender_var.get(),
        self.contact_var.get(),
        self.customer_id_var.get()
    ))
    mydb.commit()
    self.fetch_data()
    messagebox.showinfo("Update","Record has been updated.")
    self.clear()
    mydb.close()
def delete_data(self):
    mydb=sql.connect(host="localhost",user="root",password="12345",
database="saloon")#connection to mysql
    mycur=mydb.cursor()
    query="delete from customerdata where customer_id=%s"
    value=(self.customer_id_var.get(),)
    mycur.execute(query,value)
    #mycur.execute("delete from customerdata where
customer_id=%s;",self.customer_id_var.get())
    mydb.commit()
    mydb.close()
    self.fetch_data()
    messagebox.showinfo("Delete","Record has been deleted.")

    self.clear()
def search_data(self):
    mydb=sql.connect(host="localhost",user="root",password="12345",
database="saloon")#connection to mysql
    mycur=mydb.cursor()
    mycur.execute("select * from customerdata where "+str(self.search_by.get())+" LIKE
'%" +str(self.search_txt.get())+"%'"")
    rows=mycur.fetchall()
    if len(rows)!=0:

if __name__ == '__main__':
    SplashScreen()
    root=Tk()
    ob=customer(root)

```

```
root.mainloop()
```

```
self.customer_table.delete(*self.customer_table.get_children())
```

```
    for row in rows:
```

```
        self.customer_table.insert("",END,values=row)
```

```
        mydb.commit()
```

```
mydb.close()
```

OUTPUT

The image displays two screenshots of a Java Swing application titled "Login Window".

Top Screenshot: The window features a large red title "LOGIN FORM" at the top. Below it, a pink box labeled "Login" contains two input fields: "Username" and "Password". At the bottom, a light blue box labeled "Events" contains three buttons: "Login" (green text), "Reset" (black text), and "Exit" (red text).

Bottom Screenshot: This screenshot shows the same window after a successful login. The "Username" field now contains the text "root" in red, and the "Password" field contains masked input "~~~~~" in green. A small dialog box titled "Login Successfully" is overlaid on the right side of the window. The dialog box contains a question mark icon and the text "Thanks : For using Login Form." with "Yes" and "No" buttons.

LOADING

WELCOME TO ELEGANCE

The Saloon



DESIGNED BY : AMEE DADHANIYA

12 COMMERCE

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

Search By

Customer Id	Name	Services	D.O.A	Gender	Contact
-------------	------	----------	-------	--------	---------

ADD

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

ADD

UPDATE

DELETE

CLEAR

Search By

Search

Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2005-08-12	Female	9980756734
2	Bhumi	spa	2004-05-16	Female	9537380682
3	Heney	spa	2005-01-16	Female	9978803431
4	Jeel	spa	2005-12-05	Male	9567453327
5	Raj	spa	1999-05-03	Male	9978654345

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

ADD

UPDATE

DELETE

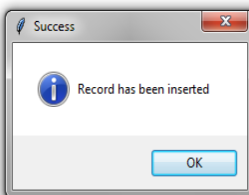
CLEAR

Search By

Search

Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2005-08-12	Female	9980756734



UPDATE

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

3

NAME

Riya

SERVICES

cut

D.O.A

2005-01-16

GENDER

Female

CONTACT

9780995654

ADD

UPDATE

DELETE

CLEAR

Search By

Search

Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2005-08-12	Female	9980756734
2	Bhumi	spa	2004-05-16	Female	9537380682
3	Riya	cut	2005-01-16	Female	9780995654
4	Jeel	spa	2005-12-05	Male	9567453327
5	Raj	spa	1999-05-03	Male	9978654345

Update

Record has been updated.

OK

DELETE

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

ADD UPDATE DELETE CLEAR

Search By Search Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2005-08-12	Female	9980756734
3	Riya	cut	2005-01-16	Female	9780995654
4	Jeel	spa	2005-12-05	Male	9567453327
5	Raj	spa	1999-05-03	Male	9978654345

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

ADD UPDATE DELETE CLEAR

Search By Search Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2005-08-12	Female	9980756734
3	Riya	cut	2005-01-16	Female	9780995654
4	Jeel	spa	2005-12-05	Male	9567453327
5	Raj	spa	1999-05-03	Male	9978654345

Delete

Record has been deleted.

OK

CLEAR

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

Search By

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2005-08-12	Female	9980756734
3	Riya	cut	2005-01-16	Female	9780995654
4	Jeel	spa	2005-12-05	Male	9567453327
5	Raj	spa	1999-05-03	Male	9978654345

Windows taskbar: 1:52 PM 12/5/2022

SEARCH BY: Customer ID

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

Search By **customer_id**

Customer Id	Name	Services	D.O.A	Gender	Contact
1	Isha	cut	2000-07-08	Female	4534556467

Windows taskbar: 10:19 AM 12/6/2022

SEARCH BY: Customer Name

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

ADD UPDATE DELETE CLEAR

Search By **name** Search Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
3	Riya	cut	2005-01-16	Female	9987764533

10:25 AM
12/6/2022

SEARCH BY: Customer Contact

ELEGANCE FAMILY SALOON

ELEGANCE CUSTOMER'S DATA

Manage Records

CUSTOMER ID

NAME

SERVICES

D.O.A

GENDER

CONTACT

ADD UPDATE DELETE CLEAR

Search By **contact** Search Show All

Customer Id	Name	Services	D.O.A	Gender	Contact
3	Riya	cut	2005-01-16	Female	9987764533

10:28 AM
12/6/2022

SEARCH BY: Services

Customer Id	Name	Services	D.O.A	Gender	Contact
4	Jeel	spa	2005-12-05	Male	8987656456
5	Raj	spa	2006-09-15	Male	8987656456

DATA STRUCTURE OUTPUT

```
mysql> use saloon;
Database changed
mysql> desc customerdata;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| customer_id | int(11)       | NO   | PRI | NULL    |       |
| name        | varchar(30)   | YES  |     | NULL    |       |
| services    | char(3)       | YES  |     | NULL    |       |
| doa         | date          | YES  |     | NULL    |       |
| gender      | varchar(10)   | YES  |     | NULL    |       |
| contact     | varchar(12)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select*from customerdata;
+-----+-----+-----+-----+-----+-----+
| customer_id | name  | services | doa       | gender | contact |
+-----+-----+-----+-----+-----+-----+
| 1           | Isha  | cut      | 2022-11-09 | Female | 9876543210 |
| 2           | Riya  | spa      | 2022-12-09 | Female | 8974563210 |
| 3           | Jeel  | cut      | 2022-10-05 | Female | 7654321890 |
| 4           | Raj   | spa      | 2022-04-06 | Male   | 9978804543 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

BIBLIOGRAPHY

**1. Computer science With Python - Class XII By
:Sumita Arora**

2. Website: <https://www.pythonworld.com>

3. Website: <https://www.xiipython.blogspot.com>

