

Python Programming (Mini Project)

Bank Management System



Project by:

Nidhi Poojary – 16010421082
Divyam Shah – 16010421095

Mini Project – Programming Laboratory 1

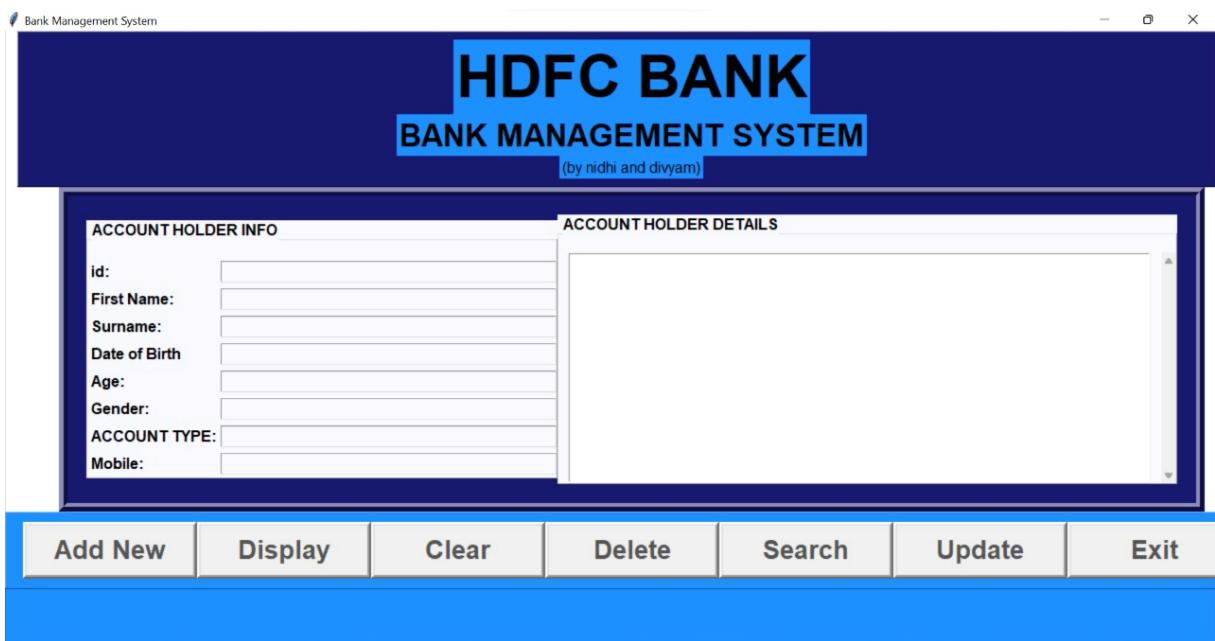
Problem Statement - The Bank Management System is a Python program where you can enter and store the details of customers of the bank. The system's goal is to ensure smooth and efficient storage of data in the bank's database. Tkinter and mysql.connector were the modules used in the project's construction. The user must enter the customer id, customer's name, phone number, dob and gender in addition to the bank account type. After entering the details, the user will click the add new button to enter the details in the database, thus displaying it in the window on the right. The user can also search, update and delete records in the database.

Before using the program, the user must have three libraries installed using ‘pip’.

- pip install tkinter
- pip install mysql.connector
- pip install messagebox

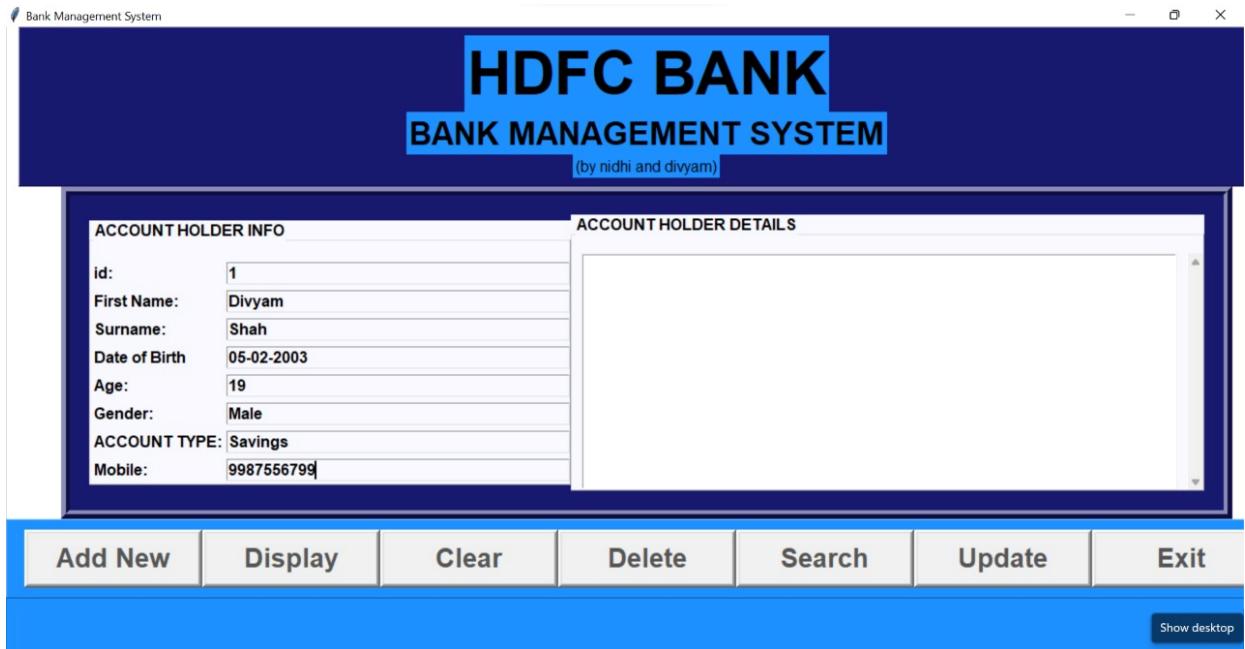
After running the three commands, we're good to go!

- When we first run the code, we see this page:



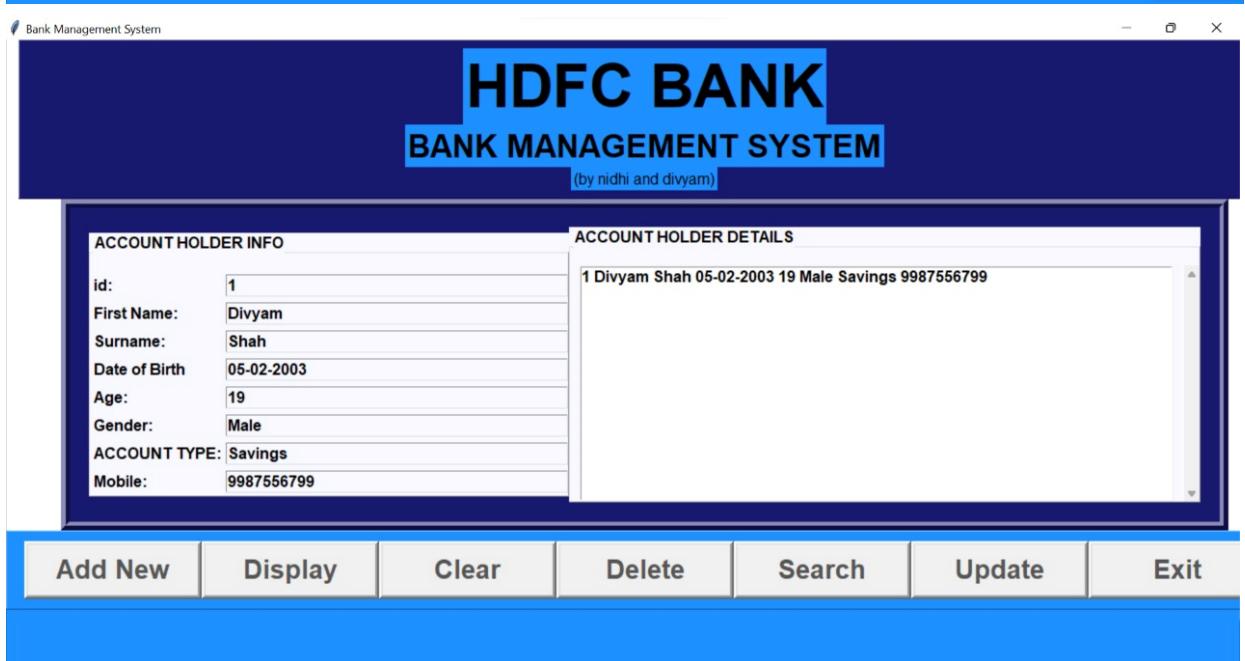
- In this page, the title, HDFC BANK(by Nidhi and Divyam), can be seen as the name of the bank. The Account Holder Info is next, where the customer's id, first name, surname, DOB, age, gender, account type and mobile number can be entered.

- Next, the records are then saved in the Account holder details window once we enter a person's information in accounts holder info window and click on "add new" button.



ACCOUNT HOLDER INFO		ACCOUNT HOLDER DETAILS	
id:	1		
First Name:	Divyam		
Surname:	Shah		
Date of Birth	05-02-2003		
Age:	19		
Gender:	Male		
ACCOUNT TYPE:	Savings		
Mobile:	9987556799		

Add New Display Clear Delete Search Update Exit



ACCOUNT HOLDER INFO		ACCOUNT HOLDER DETAILS	
id:	1	1 Divyam Shah 05-02-2003 19 Male Savings 9987556799	
First Name:	Divyam		
Surname:	Shah		
Date of Birth	05-02-2003		
Age:	19		
Gender:	Male		
ACCOUNT TYPE:	Savings		
Mobile:	9987556799		

Add New Display Clear Delete Search Update Exit

- We can insert more records in a similar way.

Bank Management System

HDFC BANK

BANK MANAGEMENT SYSTEM

(by nidhi and divyam)

ACCOUNT HOLDER INFO		ACCOUNT HOLDER DETAILS
id:	2	2 Nidhi Poojary 12-11-2003 19 Female Checkings 9987789965
First Name:	Nidhi	
Surname:	Poojary	
Date of Birth	12-11-2003	
Age:	19	
Gender:	Female	
ACCOUNT TYPE:	Checkings	
Mobile:	9987789965	

Add New | Display | Clear | Delete | Search | Update | Exit

Bank Management System

HDFC BANK

BANK MANAGEMENT SYSTEM

(by nidhi and divyam)

ACCOUNT HOLDER INFO		ACCOUNT HOLDER DETAILS
id:	3	3 Rohan Pillai 10-12-2001 21 Male Savings 9986654443
First Name:	Rohan	
Surname:	Pillai	
Date of Birth	10-12-2001	
Age:	21	
Gender:	Male	
ACCOUNT TYPE:	Savings	
Mobile:	9986654443	

Add New | Display | Clear | Delete | Search | Update | Exit

Bank Management System

HDFC BANK

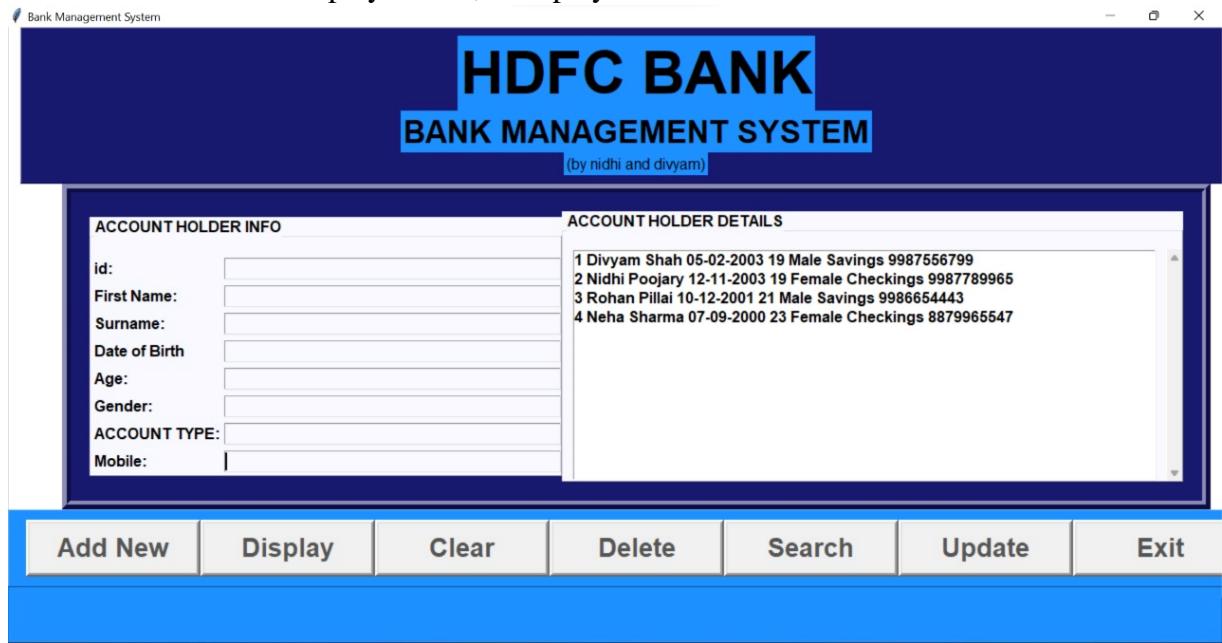
BANK MANAGEMENT SYSTEM

(by nidhi and divyam)

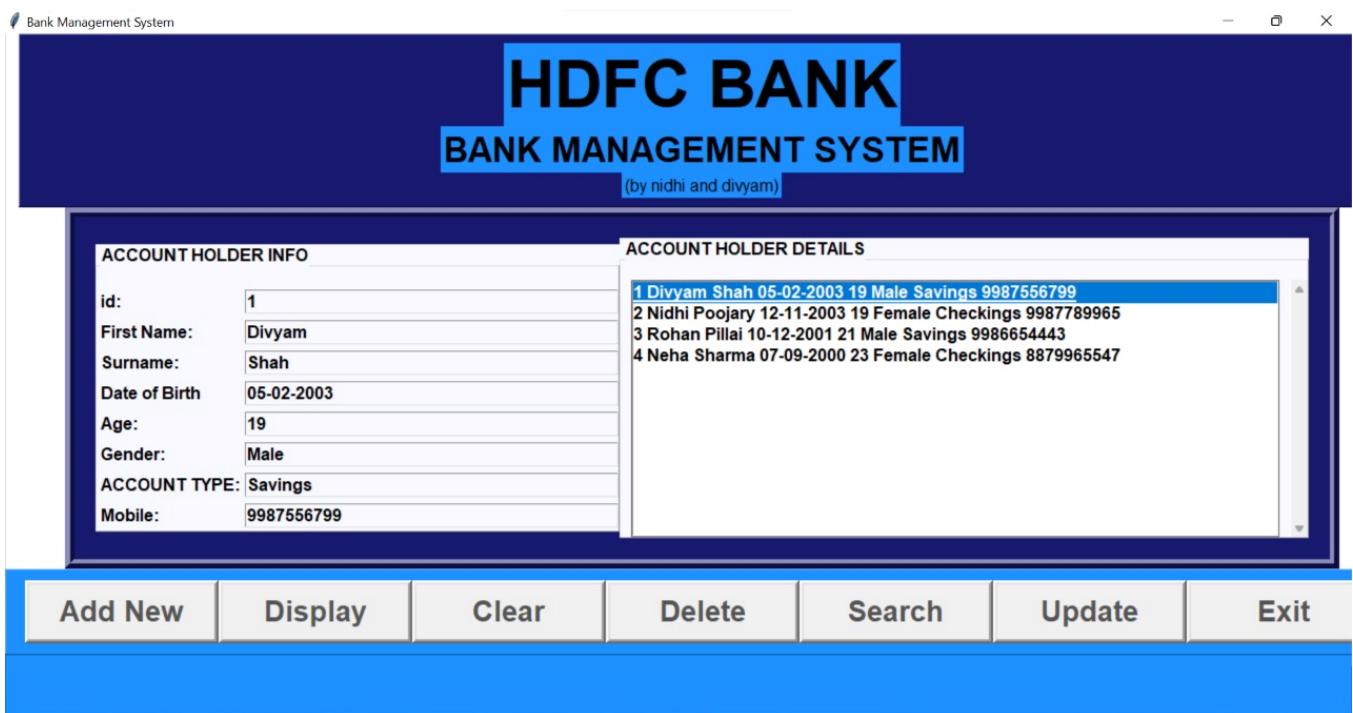
ACCOUNT HOLDER INFO		ACCOUNT HOLDER DETAILS
id:	4	4 Neha Sharma 07-09-2000 23 Female Checkings 8879965547
First Name:	Neha	
Surname:	Sharma	
Date of Birth	07-09-2000	
Age:	23	
Gender:	Female	
ACCOUNT TYPE:	Checkings	
Mobile:	8879965547	

Add New | Display | Clear | Delete | Search | Update | Exit

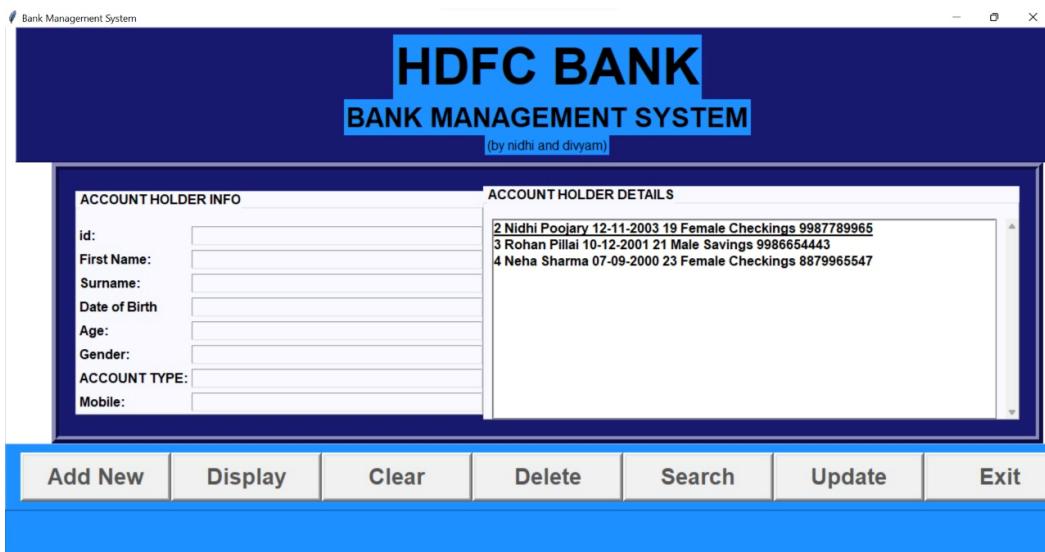
- When we click on the display button, it displays all the records entered in the window available.



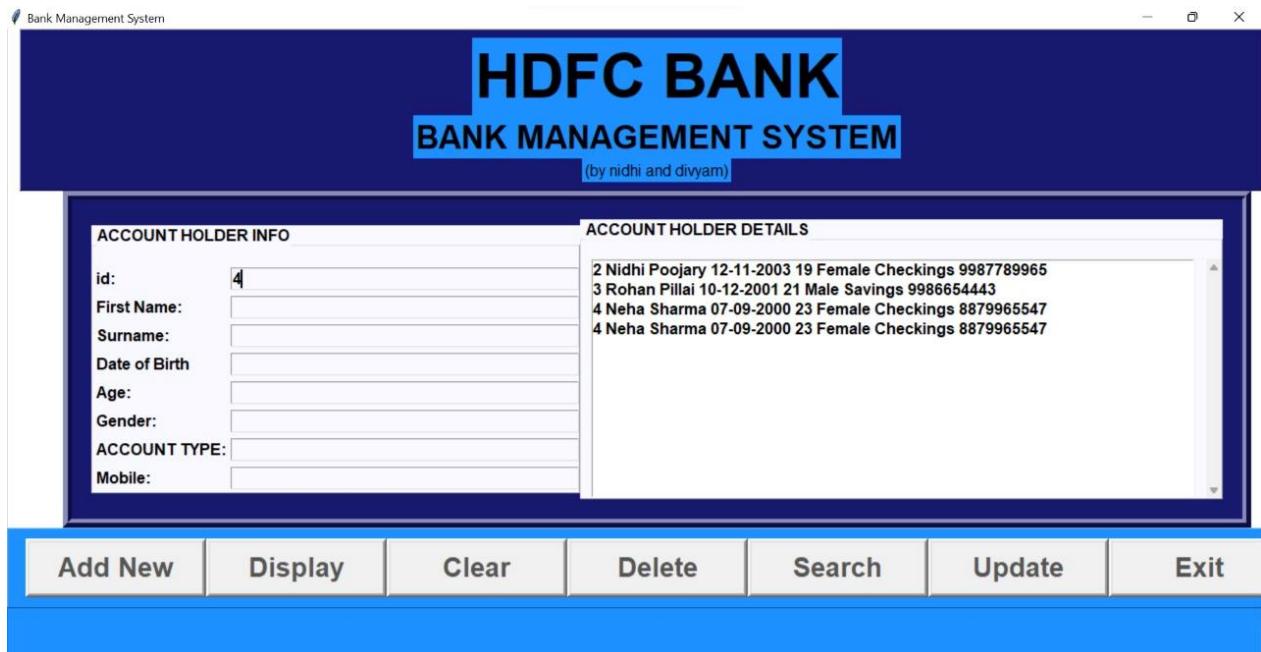
- To delete a record, we select a particular record which needs to be deleted and then click on delete button.



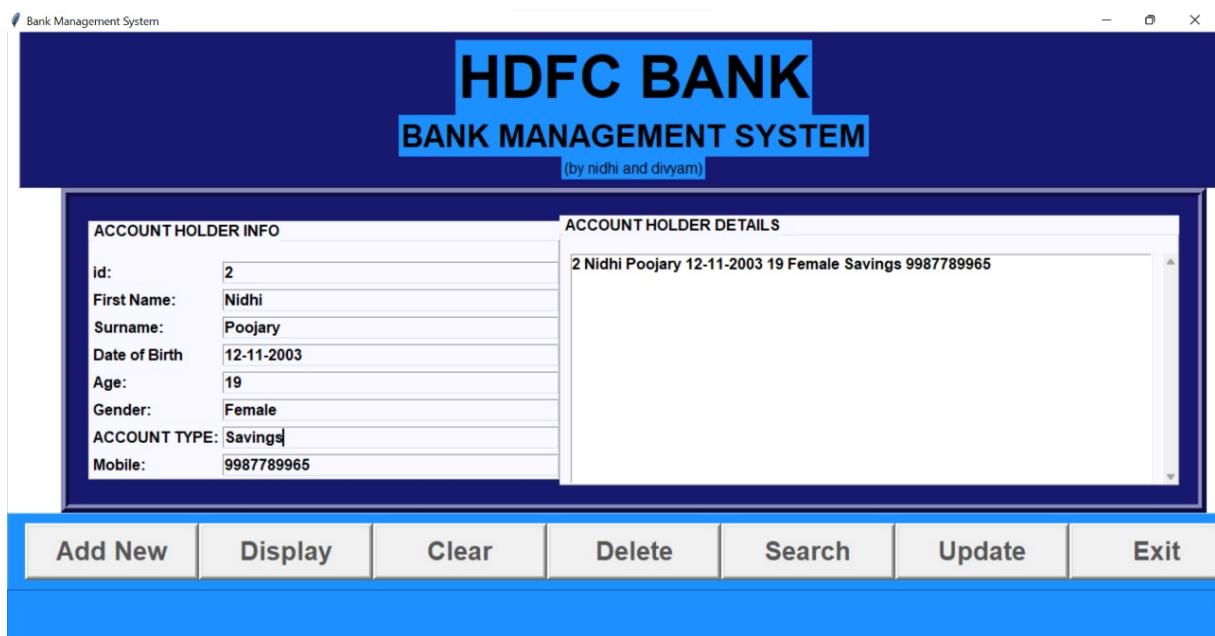
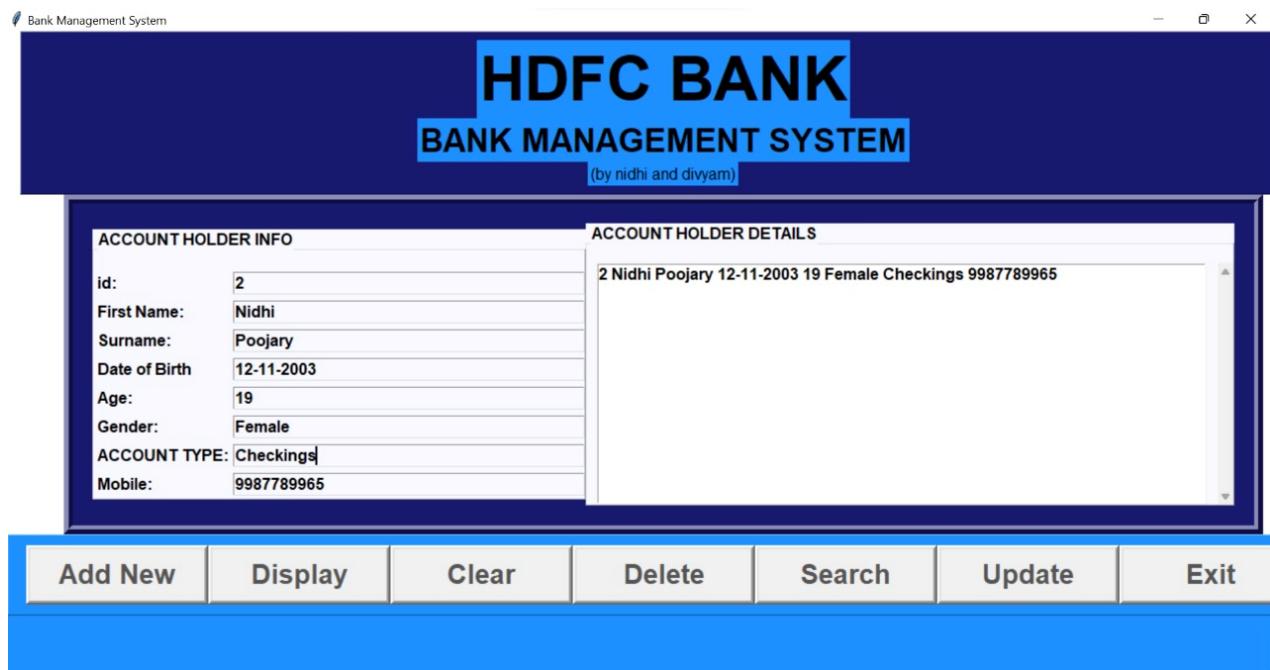
- Here we can see that the selected record has been deleted



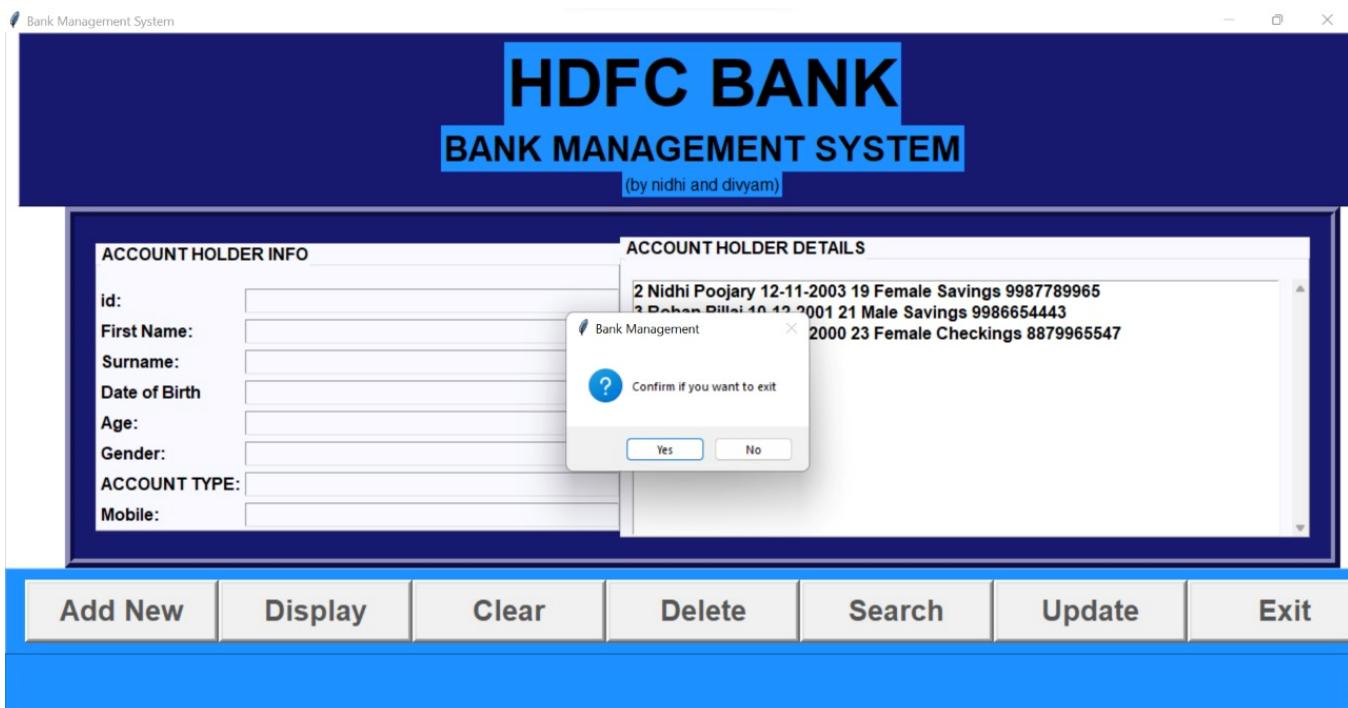
- To search a specific record, enter its Id in the Id row and the desired record will be shown at the bottom of all the records.



- We can also update a record using update button. First we need to select the record which we want to update and then change the desired fields in the account holder info window and thus the record will be updated.



- Lastly, by clicking Exit button on the GUI, the application stops and ends.



- The data entered is thus stored and is displayed in sql command line client as the program is linked to the database since we have used mysql.connector module.

```
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 226
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use project;
Database changed
mysql> show tables;
+-----+
| Tables_in_project |
+-----+
| bank
| school_management |
+-----+
2 rows in set (0.07 sec)

mysql> drop table bank;
Query OK, 0 rows affected (8.43 sec)

mysql> select * from bank;
+-----+
| ACC_ID | Firstname | Surname | DoB      | Age | Gender | ACC_TYPE | Mobile   |
+-----+
|     2  | Nidhi     | Poojary  | 12-11-2003 | 19  | Female  | Savings   | 9987789965 |
|     3  | Rohan     | Pillai   | 10-12-2001  | 21  | Male    | Savings   | 9986654443 |
|     4  | Neha      | Sharma   | 07-09-2000  | 23  | Female  | Checkings | 8879965547 |
+-----+
3 rows in set (0.00 sec)

mysql>
```

Code:

Frontend file:

```
from tkinter import *
import tkinter.messagebox
import project1_backend3 as pb

class Bank:

    def __init__(self,root):
        self.root=root
        self.root.title("Bank Management System")
        self.root.geometry(newGeometry="1328x585+0+0")
        self.root.config(bg="dodger blue")
        #ASSIGN SOME VARIABLE TO STORE OUR ENTRY FILED VALUES
        ACC_ID=StringVar()
        Firstname=StringVar()
        Surname=StringVar()
        DoB=StringVar()
        Age=StringVar()
        Gender=StringVar()
        ACC_TYPE=StringVar()
        Mobile=StringVar()
        #####FUNCTIONS#####
        pb.bankData()
    def iExit():
        iExit=tkinter.messagebox.askyesno("Bank Management","Confirm if you want to exit")
        if iExit>0:
            root.destroy()
            return
    def clearData():
        self.txtACC_ID.delete(0,END)
        self.txtFirstname.delete(0,END)
        self.txtSurname.delete(0,END)
        self.txtDob.delete(0,END)
        self.txtAge.delete(0,END)
        self.txtGender.delete(0,END)
        self.txtACC_TYPE.delete(0,END)
        self.txtMobile.delete(0,END)
    pb.bankData()
    def addData():
        if(len(ACC_ID.get())!=0):

            pb.addbankRec(ACC_ID.get(),Firstname.get(),Surname.get(),DoB.get(),Age.get(),Gender.get(),ACC_TYPE.get(),Mobile
e.get())
            studentlist.delete(0,END)
```

```

studentlist.insert(END,(ACC_ID.get(),Firstname.get(),Surname.get(),DoB.get(),Age.get(),Gender.get(),ACC_TYPE.get(),Mobile.get()))

def DisplayData():
    studentlist.delete(0,END)
    for row in pb.viewData():
        studentlist.insert(END,row)

def StudentRec(event):
    global sd
    searchstd = studentlist.curselection()[0]
    sd=studentlist.get(searchstd)
    self.txtACC_ID.delete(0,END)
    self.txtACC_ID.insert(END,sd[0])
    self.txtFirstname.delete(0,END)
    self.txtFirstname.insert(END,sd[1])
    self.txtSurname.delete(0,END)
    self.txtSurname.insert(END,sd[2])
    self.txtDob.delete(0,END)
    self.txtDob.insert(END,sd[3])
    self.txtAge.delete(0,END)
    self.txtAge.insert(END,sd[4])
    self.txtGender.delete(0,END)
    self.txtGender.insert(END,sd[5])
    self.txtACC_TYPE.delete(0,END)
    self.txtACC_TYPE.insert(END,sd[6])
    self.txtMobile.delete(0,END)
    self.txtMobile.insert(END,sd[7])

def DeleteData():
    if(len(ACC_ID.get())!=0):
        pb.deleteRec(sd[0])
        clearData()
        DisplayData()

def searchData():
    for row in
pb.searchData(ACC_ID.get(),Firstname.get(),Surname.get(),DoB.get(),Age.get(),Gender.get(),ACC_TYPE.get(),Mobile.get()):
    studentlist.insert(END,row,str(""))

def update():
    if(len(ACC_ID.get())!=0):
        pb.deleteRec(sd[0])
    if(len(ACC_ID.get())!=0):
        pb.addbankRec(ACC_ID.get(),Firstname.get(),Surname.get(),DoB.get(),Age.get(),Gender.get(),ACC_TYPE.get(),Mobile.get())
        studentlist.delete(0,END)
        studentlist.insert(END,(ACC_ID.get(),Firstname.get(),Surname.get(),DoB.get(),Age.get(),Gender.get(),ACC_TYPE.get(),Mobile.get()))

#####
MainFrame=Frame(self.root,bg="white")
MainFrame.grid() #THIS IS MAIN FRAME OUR WINDOW
TitFrame=Frame(MainFrame,bd=1,padx=400,pady=8,bg="midnight blue",relief=RIDGE)
TitFrame.pack(side=TOP)#THIS IS STITLE FRAME

self.lblTit=Label(TitFrame,font=('arial',47,'bold'),text="HDFC BANK",bg="DODGER BLUE",fg="black")
self.lblTit.grid()

```

```

self.lblTit=Label(TitFrame,font=('arial',25,'bold'),text="BANK MANAGEMENT SYSTEM",bg="DODGER BLUE",fg="black")
self.lblTit.grid()

self.lblTit=Label(TitFrame,font=('arial',12),text="(by nidhi and divyam)",bg="dodger blue",fg="black")
self.lblTit.grid()

ButtonFrame=Frame(MainFrame,bd=1,width=1350,height=70,padx=18,pady=10,bg="dodger blue",relief=RIDGE)
ButtonFrame.pack(side=BOTTOM)#

DataFrame=Frame(MainFrame,bd=9,width=1300,height=400,padx=20,pady=20,bg="midnight blue",relief=RIDGE)
DataFrame.pack(side=BOTTOM)#THIS IS STI

DataFrameLeft=LabelFrame(DataFrame,font=('arial',12,'bold'),bd=1,width=450,height=300,bg="Ghost
White",relief=RIDGE,text="ACCOUNT HOLDER INFO\n")
DataFrameLeft.pack(side=LEFT)

DataFrameRight=LabelFrame(DataFrame,font=('arial',12,'bold'),bd=1,width=450,height=300,bg="Ghost
White",relief=RIDGE,text="ACCOUNT HOLDER DETAILS\n")
DataFrameRight.pack(side=RIGHT)

#####Labels and entry widget#####

self.lblACC_ID=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="id:",bg="ghost white")
self.lblACC_ID.grid(row=0,column=0,sticky=W)

self.txtACC_ID=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=ACC_ID,bg="ghost white",width=39)
self.txtACC_ID.grid(row=0,column=1)#id

self.lblFirstname=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="First Name:",bg="ghost white")
self.lblFirstname.grid(row=1,column=0,sticky=W)

self.txtFirstname=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=Firstname,bg="ghost white",width=39)
self.txtFirstname.grid(row=1,column=1)#firstname

self.lblSurname=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="Surname:",bg="ghost white")
self.lblSurname.grid(row=2,column=0,sticky=W)

self.txtSurname=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=Surname,bg="ghost white",width=39)
self.txtSurname.grid(row=2,column=1)#surname

self.lblDob=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="Date of Birth",bg="ghost white")
self.lblDob.grid(row=3,column=0,sticky=W)

self.txtDob=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=DoB,bg="ghost white",width=39)
self.txtDob.grid(row=3,column=1)#dateof birth

self.lblAge=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="Age:",bg="ghost white")
self.lblAge.grid(row=4,column=0,sticky=W)

self.txtAge=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=Age,bg="ghost white",width=39)
self.txtAge.grid(row=4,column=1)#age

self.lblGender=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="Gender:",bg="ghost white")
self.lblGender.grid(row=5,column=0,sticky=W)

self.txtGender=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=Gender,bg="ghost white",width=39)

```

```

self.txtGender.grid(row=5,column=1)#gender

self.lblACC_TYPE=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="ACCOUNT TYPE:",bg="ghost white")
self.lblACC_TYPE.grid(row=6,column=0,sticky=W)

self.txtACC_TYPE=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=ACC_TYPE, bg="ghost white",width=39)
self.txtACC_TYPE.grid(row=6,column=1)#TYPE

self.lblMobile=Label(DataFrameLeft,font=('arial',12,'bold'),padx=2,pady=3,text="Mobile:",bg="ghost white")
self.lblMobile.grid(row=7,column=0,sticky=W)

self.txtMobile=Entry(DataFrameLeft,font=('arial',12,'bold'),textvariable=Mobile, bg="ghost white",width=39)
self.txtMobile.grid(row=7,column=1)#mobile

#####List Box and ScrollBar Widget#####
scrollbar=Scrollbar(DataFrameRight)
scrollbar.grid(row=0 ,column=1,sticky='ns')#scroll bar

studentlist=Listbox(DataFrameRight,width=68,height=12,font=('arial',12,'bold'), yscrollcommand=scrollbar.set)
studentlist.bind('<<ListboxSelect>>',StudentRec)
studentlist.grid(row=0,column=0,padx=10)
scrollbar.config(command= studentlist.yview)

#####Button Widget#####
self.btnAddData=Button(ButtonFrame,text="Add New",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=addData)
self.btnAddData.grid(row=0,column=0)#ADD NEW

self.btnDisplay=Button(ButtonFrame,text="Display",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=DisplayD
ata)
self.btnDisplay.grid(row=0,column=1)#DISPLAY

self.btnClear=Button(ButtonFrame,text="Clear",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=clearData)
self.btnClear.grid(row=0,column=2)#CLEAR

self.btnDelete=Button(ButtonFrame,text="Delete",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=DeleteData
)
self.btnDelete.grid(row=0,column=3)#DELETE

self.btnSearch=Button(ButtonFrame,text="Search",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=searchData
base)
self.btnSearch.grid(row=0,column=4)#SEARCH

self.btnUpdate=Button(ButtonFrame,text="Update",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=update)
self.btnUpdate.grid(row=0,column=5)#UPDATE

self.btnExit=Button(ButtonFrame,text="Exit",font=('arial',20,'bold'),height=1,width=10,bd=4,fg="#555",command=iExit)
self.btnExit.grid(row=0,column=6)#EXIT

if __name__=='__main__':
    root=Tk()#CREATE AN OBJECT
    application=Bank(root)#PASS IT TO OUR CLASS WHITH ITS PROPERTIES IN CLASS
    root.mainloop()#RUN UNTIL CLOSING THE WINDOW MANUALLY

```

Backend file:

```

import mysql.connector
#import project1_frontend

def bankData():
    con = mysql.connector.connect(host="localhost",user="root",passwd="Password12*",database="project")
    cur=con.cursor()

    cur.execute("CREATE TABLE IF NOT EXISTS Bank(ACC_ID integer primary key AUTO_INCREMENT,Firstname text,Surname text,DoB text,Age text,Gender text,ACC_TYPE text,Mobile text)")
    con.commit()
    con.close()

def addbankRec(ACC_ID,Firstname,Surname,DoB,Age,Gender,ACC_TYPE,Mobile):
    con=con = mysql.connector.connect(host="localhost",user="root",passwd="Password12*")
    cur=con.cursor()
    cur.execute("use project")
    cur.execute("INSERT INTO Bank VALUES(%s,%s,%s,%s,%s,%s,%s)",(ACC_ID,Firstname,Surname,DoB,Age,Gender,ACC_TYPE,Mobile))
    con.commit()
    con.close()

def viewData():
    con=con = mysql.connector.connect(host="localhost",user="root",passwd="Password12*")
    cur=con.cursor()
    cur.execute("use project")
    cur.execute("select * from Bank")
    row=cur.fetchall()
    con.close()
    return row

def deleteRec(ACC_ID):
    con=con = mysql.connector.connect(host="localhost",user="root",passwd="Password12*")
    cur=con.cursor()
    cur.execute("use project")
    cur.execute("DELETE FROM Bank WHERE ACC_ID=%s",(ACC_ID,))
    con.commit()
    con.close()

def searchData(ACC_ID,Firstname,Surname,DoB,Age,Gender,Mobile,ACC_TYPE):
    con=con = mysql.connector.connect(host="localhost",user="root",passwd="Password12*")
    cur=con.cursor()
    cur.execute("use project")
    cur.execute("SELECT * FROM Bank WHERE ACC_ID=%s or Firstname=%s or Surname=%s or DoB=%s or Age=%s or Gender=%s or ACC_TYPE=%s or Mobile=%s",(ACC_ID,Firstname,Surname,DoB,Age,Gender,ACC_TYPE,Mobile))
    rows=cur.fetchall()
    con.close()
    return rows

def dataUpdate(ACC_ID="",Firstname:"",Surname:"",DoB:"",Age:"",Gender:"",ACC_TYPE:"",Mobile:""):
    con = mysql.connector.connect(host="localhost",user="root",passwd="Password12*")
    cur=con.cursor()
    cur.execute("use project")
    cur.execute("UPDATE Bank SET ACC_ID=%s,Firstname=%s,Surname=%s,DoB=%s,Age=%s,Gender=%s,ACC_TYPE=%s,Mobile=%s WHERE ACC_ID=%s",(ACC_ID,Firstname,Surname,DoB,Age,Gender,ACC_TYPE,Mobile))
    con.commit()
    con.close()

```

Conclusion:

As a result of this project, we were able to improve our skills in the area of creating a Graphical User Interface (GUI) using the tkinter library. We also learned how to create a bank management system using modules from tkinter and mysql connector in python.

References:

<https://realpython.com/python-gui-tkinter/>

<https://www.javatpoint.com/python-tkinter>

https://www.w3schools.com/python/python_mysql_getstarted.asp

<https://realpython.com/python-mysql/>