ITAPA2-B12 Project 2

Index:

Section A:

Question 1:

```
def Choice(Continue): #create the choice function to give an user the choice to restart the roman converter
    if Continue == "Yes":
        RomanConvert() #Calls the romanconvert program to restart it
    else:
        print("Thank you for using this program!")
        exit() #Exits the program

RomanDict = {'i':1,
             'v':5,
             'x':10,
             'l':50,
             'c':100,
             'd':500,
             'm':1000
            } #Save the roman values into a dictionary
def RomanConvert():
    Value = 0 #The total roman value is still undertermined and stays at 0
    Roman = input("Enter Roman numerals to be converted into normal numbers") #Prompts for a roman value to be converted
    for X in range(len(Roman)): #Loops through the whole value entered by the user and converts every letter it gets
        if X+1 != len(Roman) and RomanDict[Roman[X]]<RomanDict[Roman[X+1]]:
            Value = Value - RomanDict[Roman[X]]
            print(Value) #Display the converted roman value into normal numerals
        else:
            Value = Value + RomanDict[Roman[X]]
            print(Value)
    Retry= input("Would you like to try again?") #prompts the user if they would like to run the program again
    Choice(Retry) #Calls the Choice function to either restart the program or exit it

RomanConvert()
```

```
Enter Roman numerals to be converted into normal numbersiv
-1
4
Would you like to try again?Yes
Enter Roman numerals to be converted into normal numbersixxxx
-1
9
19
29
39
Would you like to try again?dssadd
Thank you for using this program!
```

Question 2:

```
def Deposit(Money): #Create the deposit function to deposit money
    Amount = int(input("Please enter the amount of money to deposit into your account")) #Enter how much money to deposit
    Money = Money + Amount #The deposited money gets added to the total balance
    print("Your new balance is : ", Money) #Displays the total Money left in the account
    bank.Choice() #Prompts the user if they want to do anything else on their bank account

def Withdraw(Money): #Create the withdraw function to withdraw money
    Withdraw = int(input("Please enter the amount of money to withdraw from your account")) #Enter how much money to withdraw
    Money = Money - Withdraw #Total amount of money in account is decreased by the amount withdrawn by the user
    print("Withdrawal processed")
    print("Your new balance is : ", Money) #Displays the total Money left in the account
    bank.Choice()

def Total(Money): #Create the Total function to view the total money of a user
    print("You have", Money, "rand on your account") #Displays the total Money left in the account
    bank.Choice()

def Change_Pin(PIN): #Create the change pin function to change a password
    newpin1 = input("Please enter a new pin") #Asks for the user to enter a new pin
    newpin2 = input("Please confirm your new pin") #Asks for the user to confirm their new pin
    while newpin2 != newpin1: #Tests if both pins entered match
        print("Wrong you must repeat your new pin")
        newpin2 = input("Please confirm your new pin")
    PIN = newpin1 #Sets the old password to the new password
    print("New pin saved")
    bank.Choice()

class Bank: #Create the Bank Class to create the main ATM
    def _init_(self, User, PIN, Money): #Initialize the class
        self.User = User #Create the variables that will be used throughout the program
        self.PIN = PIN
        self.Money = Money
```

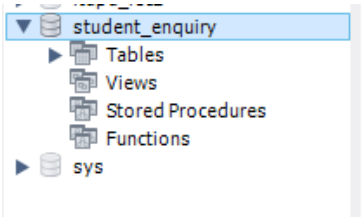
```
    def Login(self): #Create the Login function to ask for a user's login credentials
        self.User = ["user1","user2","user3"] #Create the three users
        self.PIN = ["12345", "678910", "pass123"] #Create the three passwords for the three users
        self.Money = [10000, 5000, 15000] #Create the total balance for the users
        global Username, Password, Balance
        Username = input("Please enter your Username:") #Ask for the user's username
        if Username == self.User[0]: #If the username is correct, the password can then be inserted
            Password = input("Please enter your PIN") #Ask for the user's password
            if Password == self.PIN[0]: #If the password is correct, the user can access the bank
                Balance = self.Money[0] #Set the balance to the corresponding user's money
                bank.Choice() #Call the choice function
        if Username == self.User[1]:
            Password = input("Please enter your PIN")
            if Password == self.PIN[1]:
                Balance = self.Money[1]
                bank.Choice()
        if Username == self.User[2]:
            Password = input("Please enter your PIN")
            if Password == self.PIN[2]:
                Balance = self.Money[2]
                bank.Choice()

    def Choice(self):
        print("1. Make a Deposit")
        print("2. Make a Withdrawal")
        print("3. Obtain Balance")
        print("4. Change Pin")
        print("5. Quit")
        choice = input("Please make a selection") #promts the user to select an option below
        if choice == "1":
            Deposit(Balance) #Call the deposit function
        if choice == "2":
            Withdraw(Balance) #Call the withdraw function
        if choice == "3":
            Total(Balance) #Call the Total function
        if choice == "4":
            Change_Pin(Password) #Call the change pin function
        if choice == "5":
            print("Goodbye")
            exit() #If the user choose this option, the program exits

bank = Bank() #Call the Class so that it can be created
bank.Login() #Call the Login function to start the program
```

```
Please enter your Username:user1
Please enter your PIN12345
1. Make a Deposit
2. Make a Withdrawal
3. Obtain Balance
4. Change Pin
5. Quit
Please make a selection1
Please enter the amount of money to deposit into your account5000
Your new balance is :  15000
1. Make a Deposit
2. Make a Withdrawal
3. Obtain Balance
4. Change Pin
5. Quit
Please make a selection2
Please enter the amount of money to withdraw from your account5000
Withdrawal processed
Your new balance is :  5000
1. Make a Deposit
2. Make a Withdrawal
3. Obtain Balance
4. Change Pin
5. Quit
Please make a selection3
You have 10000 rand on your account
1. Make a Deposit
2. Make a Withdrawal
3. Obtain Balance
4. Change Pin
5. Quit
Please make a selection4
Please enter a new pin6789
Please confirm your new pin6789
New pin saved
1. Make a Deposit
2. Make a Withdrawal
3. Obtain Balance
4. Change Pin
5. Quit
Please make a selection5
Goodbye
```

Question 3:

Creating the Database:

```
import mysql.connector #connecting python to mysql to use it
Student = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Thunderdrake21"
)
Cursor = Student.cursor() #Calling the cursor to use it
Cursor.execute("CREATE DATABASE Student_Enquiry") #creating the database
```



Database Created

Creating the table

```
import mysql.connector
Student = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Thunderdrake21",
    database="Student_Enquiry" #Connect the created database to python
)
Cursor = Student.cursor() #Calling the cursor to use it
Cursor.execute("CREATE TABLE Student_Info(StudentNum INT auto_increment PRIM
#Create the table with its columns by using the execute command for the curs
```

```
e created database to python

sor to use it
StudentNum INT auto_increment PRIMARY key, Name VARCHAR(40), Surname VARCHAR(40), Email VARCHAR(50), Query VARCHAR(255))")
g the execute command for the cursor
```

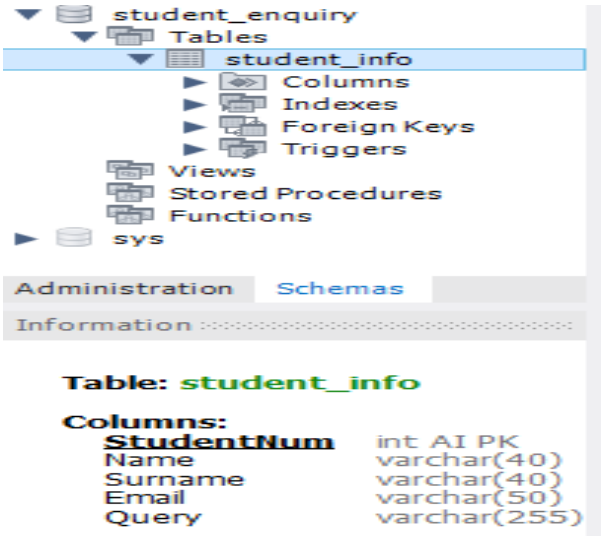


Table Created

Creating the GUI:

```
DisplayText = Text(Studentroot) #Create a Text form to display data on
DisplayText.grid(row=20, column=1) #Place the Text form on the GUI
Infolabel = Label(Studentroot, text="Enter your details to log a query") #Create a label "Enter your details to log a query"
Spacelabel1 = Label(Studentroot, text= " " ) #Create a label that displays  a space for positioning
Spacelabel2 = Label(Studentroot, text= " " ) #Create a label that displays  a space for positioning
Spacelabel3 = Label(Studentroot, text= " " ) #Create a label that displays  a space for positioning
Spacelabel4 = Label(Studentroot, text= " " ) #Create a label that displays  a space for positioning
Namelabel = Label(Studentroot, text="Name") #Create a label that displays "Name"
Surnamelabel = Label(Studentroot, text="Surname") #Create a label that displays "Surname"
Emaillabel = Label(Studentroot, text="Email") #Create a label that displays "Email"
Querylabel = Label(Studentroot, text="Student Query") #Create a label that displays "Student Query"
Submitbutton = Button(Studentroot, text = "Submit", command = logquery) #Create a button that will execute the logquery function
Searchlabel = Label(Studentroot, text="Enter a name to search") #Create a label that displays "Search"
Searchbutton = Button(Studentroot, text = "Search", command = SearchRecord)#Create a button that will execute Search function
Viewbutton = Button(Studentroot, text = "View All", command = DisplayData)#Create a button that will execute the display functior
Clearbutton = Button(Studentroot, text = "Clear Form", command = Clear) #Create a button that will execute the Clear Function
Deletebutton = Button(Studentroot, text = "Delete", command = Delete) #Create a button that will execute the Delete function
NameEntry = Entry(Studentroot) #Gives the user acces to enter a name
SurNameEntry = Entry(Studentroot) #Gives the user acces to enter a surname
EmailEntry = Entry(Studentroot) #Gives the user acces to enter an Email adress
QueryEntry = Entry(Studentroot) #Gives the user acces to enter a query
SearchEntry = Entry(Studentroot) #Gives the user acces to search for a name
Infolabel.grid(row=0, column=0) #Places the Info label on the GUI
Spacelabel1.grid(row=1, column=0) #Places the Space1 label on the GUI
Namelabel.grid(row=2, column=0) #Places the Name label on the GUI
NameEntry.grid(row=3, column=0) #Places the Name Entry on the GUI
Surnamelabel.grid(row=4, column=0) #Places the Surname label on the GUI
SurNameEntry.grid(row=5, column=0) #Places the SurName Entry on the GUI
Emaillabel.grid(row=6, column=0) #Places the Email label on the GUI
EmailEntry.grid(row=7, column=0) #Places the Email Entry on the GUI
Querylabel.grid(row=8, column=0) #Places the Query label on the GUI
QueryEntry.grid(row=9, column=0) #Places the Query Entry on the GUI
Submitbutton.grid(row=10, column=0) #Places the Submitbutton on the GUI
Searchlabel.grid(row=11, column=0) #Places the Search label on the GUI
SearchEntry.grid(row=12, column=0) #Places the Search Entry on the GUI
Searchbutton.grid(row=13, column=0) #Places the Search button on the GUI
```

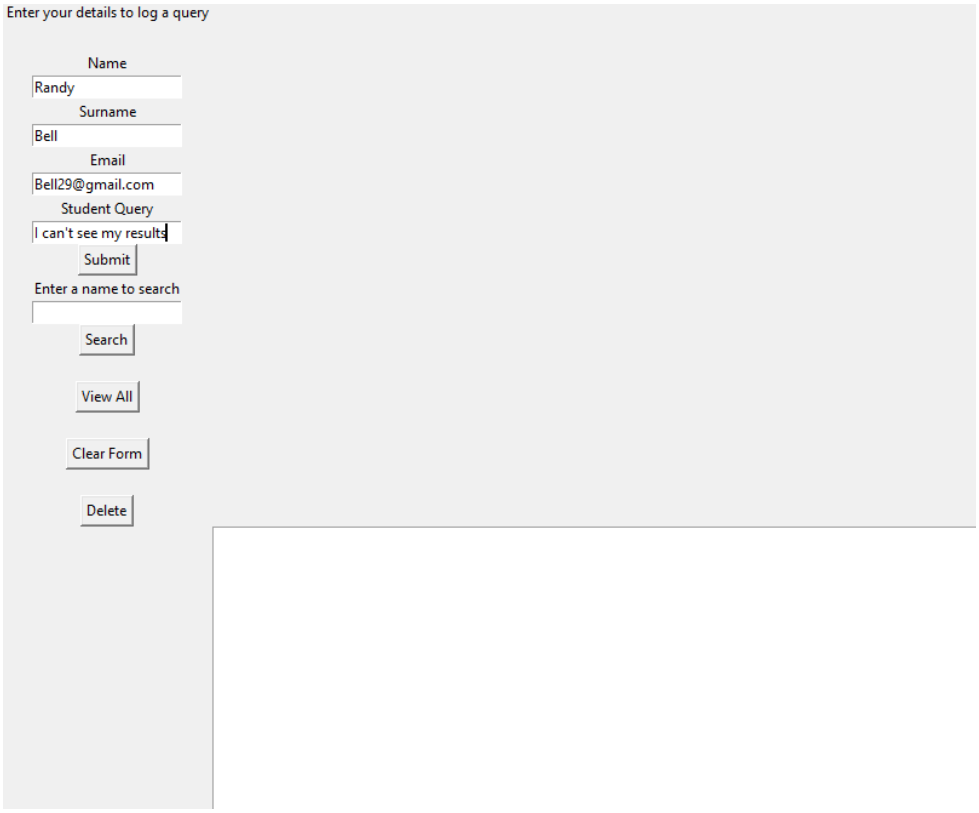
```
Spacelabel2.grid(row=14, column=0) #Places the Space2 label on the GUI
Viewbutton.grid(row=15, column=0) ##Places the View button on the GUI
Spacelabel3.grid(row=16, column=0) #Places the Space3 label on the GUI
Clearbutton.grid(row=17, column=0) #Places the Clear button on the GUI
Spacelabel4.grid(row=18, column=0) #Places the Space4 label on the GUI
Deletebutton.grid(row=19, column=0) #Places the Delete button on the GUI
Studentroot.mainloop() #Loops the GUI, until the user disables it
```

Functions:

```
from tkinter import * #Import the tkinter function to create a GUI
import mysql.connector
Studentroot = Tk() #Create the GUI
Student = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Thunderdrake21",
    database="Student_Enquiry"
)
Cursor = Student.cursor() #Calling the cursor to use it
#if submit is clicked
def logquery(): #Create the logquery function that will insert the data entered by the user into the database
    StudentName = NameEntry.get() #Gets the data inserted in NameEntry
    StudentSurname = SurNameEntry.get() #Gets the data inserted in SurnameEntry
    StudentEmail = EmailEntry.get() #Gets the data inserted in EmailEntry
    StudentQuery = QueryEntry.get() #Gets the data inserted in QueryEntry
    Insert = "INSERT INTO Student_Info(Name, Surname, Email, Query)VALUES(%s, %s, %s, %s)" #Call the INSERT function
    Input = (StudentName, StudentSurname, StudentEmail, StudentQuery) #Data that will be inserted into the database
    Cursor.execute(Insert, Input) #Executes the two functions
    Student.commit() #Insert the data entered by the user into the database
```

```
#if the delete button is clicked
def Delete(): #Create the Delete function to delete records
    StudentName = SearchEntry.get() #Search for the record to be deleted
    Delete = "Delete * from Student_Info Where Name = %s"
    Name = (StudentName, )
    Cursor.execute(Delete, Name) #Executes the delete function
    Student.commit() #Deletes the record in the database
#if the search button is clicked
def SearchRecord(): #Create the SearchRecord function to search for records
    SearchName = SearchEntry.get() #Search for a record
    Search = "Select * from Student_Info Where Name = %s"
    Name = (SearchName, )
    Cursor.execute(Search, Name) #Executes the select function
    Result = Cursor.fetchall() #Execute to select the data of the name entered
    for S in Result:
        DisplayText.insert("1.0",S) #Displays all the data of the name searched
#if the viewall button is clicked
def DisplayData(): #Create the Displaydata function to display all the records of the database
    Cursor.execute("Select * from Student_Info ORDER BY Name") #Selects everything from the database and orders it by name
    Result = Cursor.fetchall() #Fetch all the database data
    for S in Result:
        DisplayText.insert("1.0",S) #Displays Everthing from the database
#if the clear button is clicked
def Clear(): #Create the Clear function to clear the form of data
    DisplayText.delete("1.0", Studentroot.END) #Clears the form of data
```

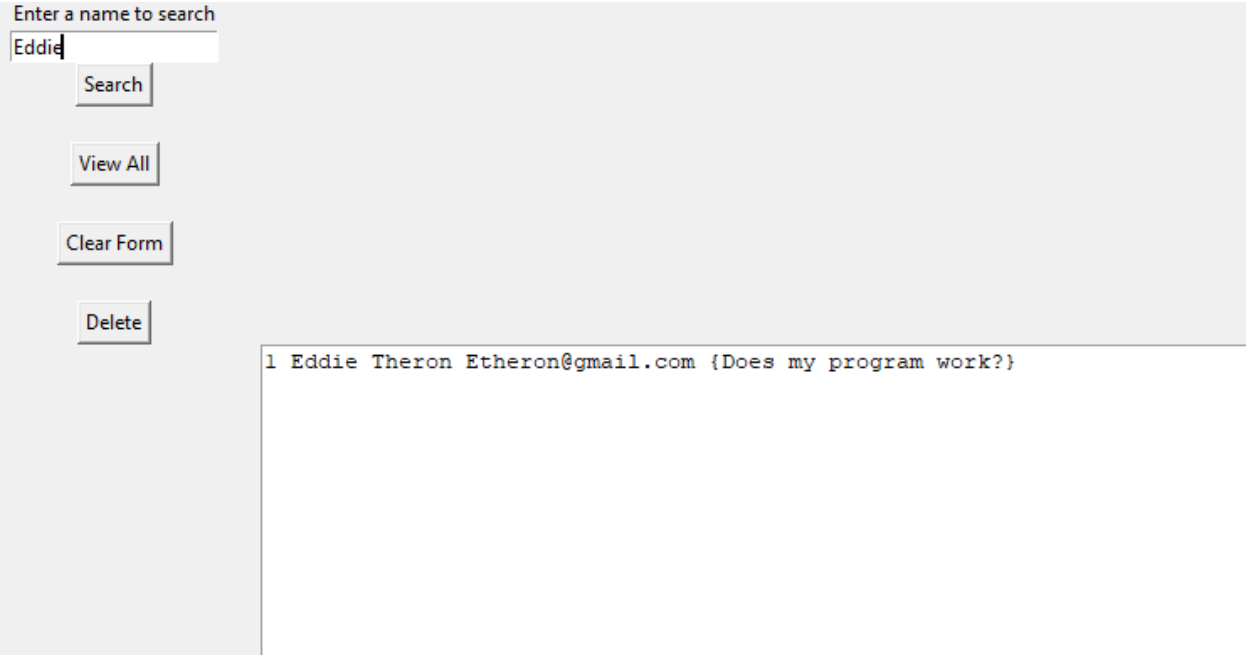
Submit Button (log query function):



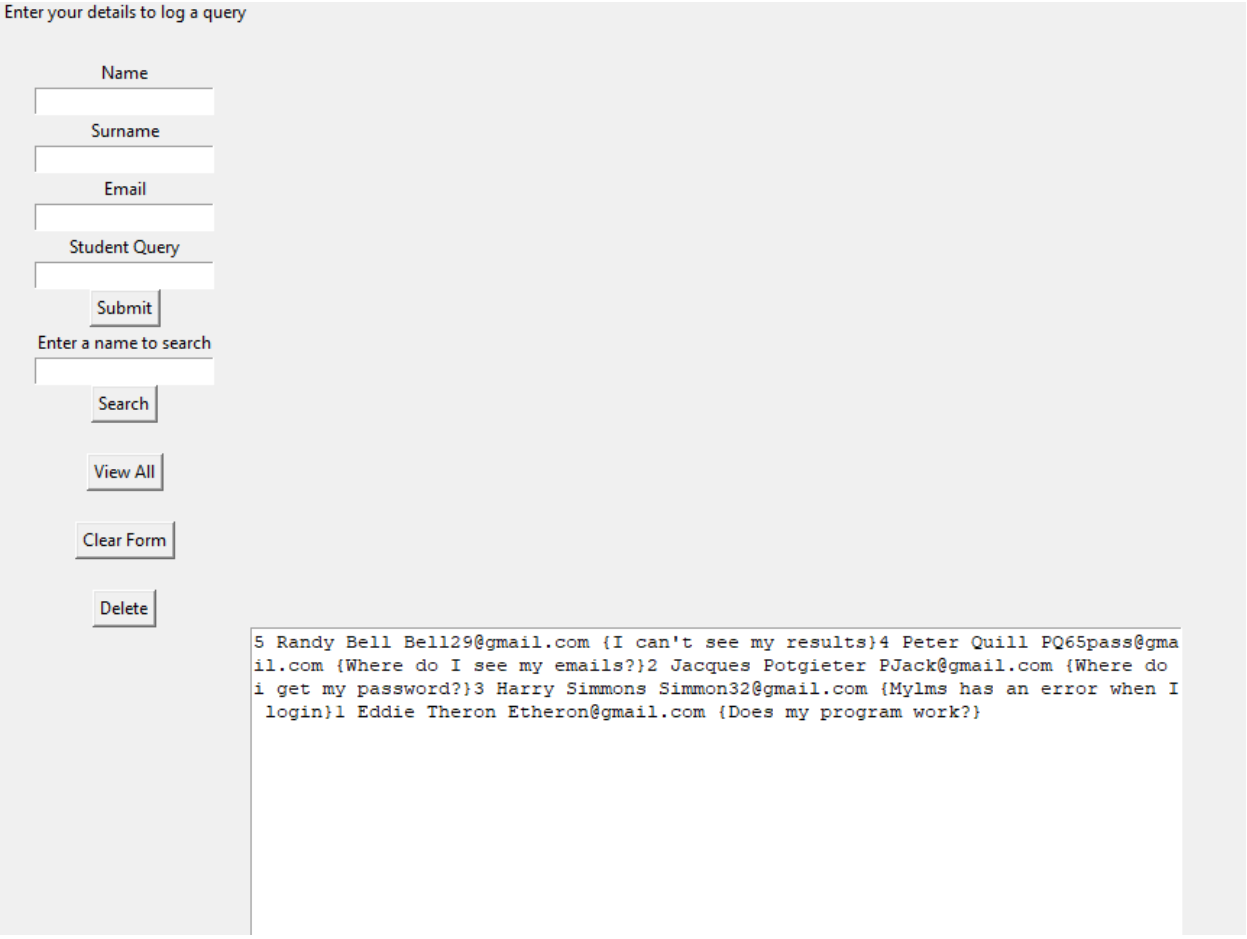
When submit is clicked

StudentNum	Name	Surname	Email	Query
1	Eddie	Theron	Etheron@gmail.com	Does my program work?
2	Jacques	Potgieter	PJack@gmail.com	Where do i get my password?
3	Harry	Simmons	Simmon32@gmail.com	Mylms has an error when I login
4	Peter	Quill	PQ65pass@gmail.com	Where do I see my emails?
5	Randy	Bell	Bell29@gmail.com	I can't see my results

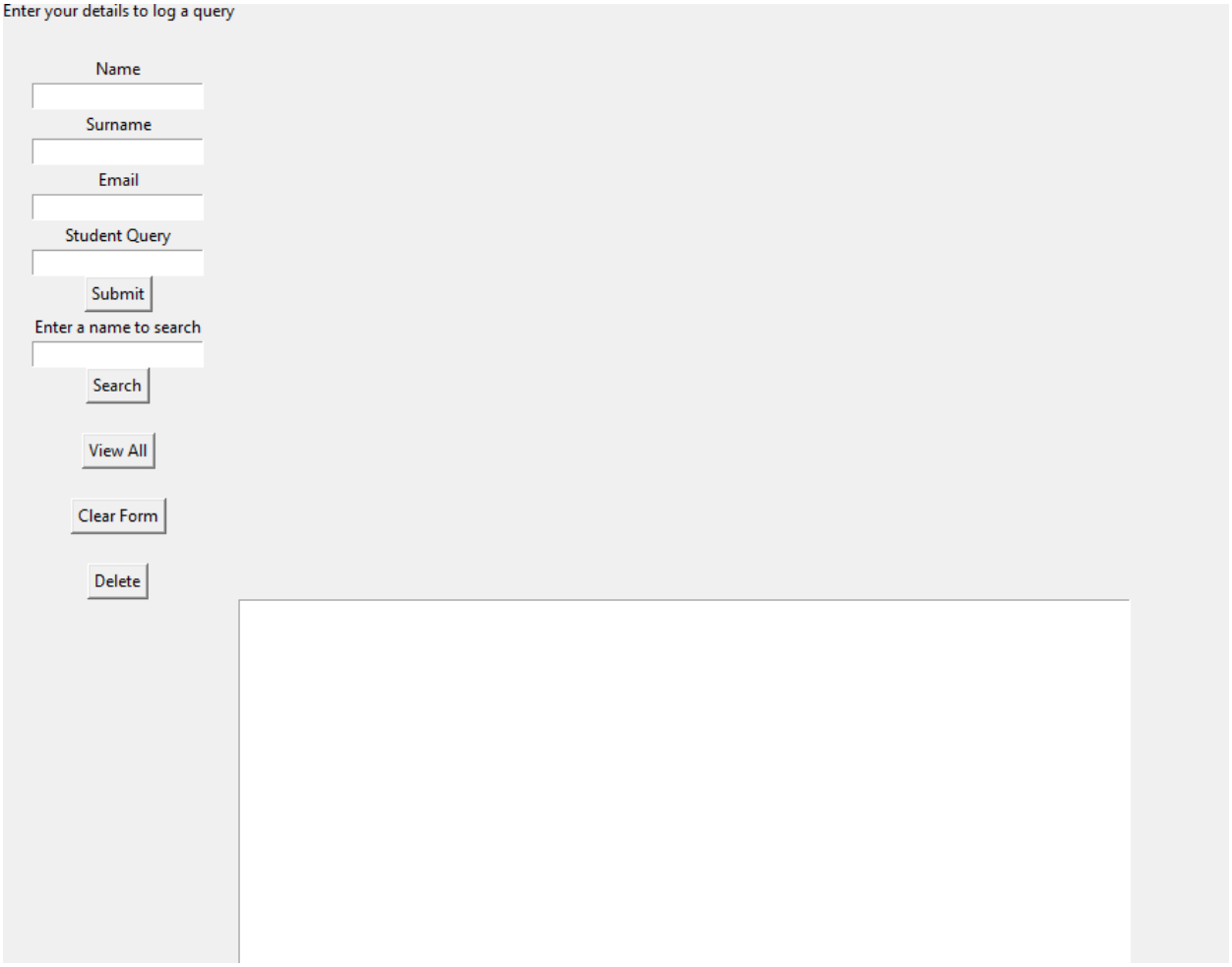
Search Button (Search Record Function):



View all Button (Display Data Function):



Clear Form Button (Clear Function):



Bibliography:

1. N/A, 2021, Python: Passing variables between variables, Stackoverflow.com
   https://stackoverflow.com/questions/16043797/python-passing-variables-between-functions
   [Date Accessed: 26 June 2021]
2. freeCodeCamp.org, 2019, Tkinter Course-Create Graphic User Interfaces in Python Tutorial, Youtube.com
   https://www.youtube.com/watch?v=YXPyB4XeYLA
   [Date Accessed: 26 June 2021]
3. Amos D, 2021, Python GUI Programming with Tkinter, realpython.com
   https://realpython.com/python-gui-tkinter/
   [Date Accessed: 28 June 2021]
4. N/A,2021, Python MySQL Delete from by, www.w3schools.com
   https://www.w3schools.com/python/python_mysql_delete.asp
   [Date Accessed: 29 June 2021]
5. Lububu S, 2021, Various PowerPoint Slides
   [Date Accessed: 26 – 30 June 2021]
6. Engineers Revolution, 2020, Program to Convert roman numerals to Numbers in Python, Youtube.com
   https://www.youtube.com/watch?v=nCDoMK-E7mc
   [Date Accessed: 30 June 2021]