**REPORT**

**ON**

**CRIMINAL RECORD**

**BY**

**Section : K17PJ**

|  |  |  |
| --- | --- | --- |
| **Registration no.** | **Name** | **Roll no.** |
| 11706739 | Saloni Jain | A-14 |
| 11711047 | Vikas Raria | B-27 |
| 11716689 | Shakti | B-29 |

**Session :** 2018~2019

**Submitted to :** Gagandeep Kaur



**School of Computer Science and Engineering**

**Lovely Professional University**

**Phagwara ,Punjab(India)**

**OBJECTIVE**

**Design a system for Criminal record with full details**

**INTRODUCTION**

The aim behind this topic is to make a unique and new project that stores the records and details of a criminal in a Jail system.

The Jail named as OXFORD JAIL which stores the records of the criminal in a table named as Criminal Record. The stored information contains the Criminal Name, Criminal number, Crime, Age, Father’s name, City, Mobile number, Time of his/her punishment, etc..

In this project a GUI interface is designed such that the new Criminals data will be added to the table, the data will be searched of the criminal by just entering his/her name, the data will also be updated as well as deleted too.

**GUI INTERFACE**

**GUI** stands for Graphical User interface that allows user to interact with digital devices or electric devices through graphical icons and visual indicators.

Python provides various options for developing graphical user interfaces (GUIs).

**TKINTER**

Python provides various options for developing graphical user interfaces (GUIs).

Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

* Import the Tkinter module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user.

**FOR EXAMPLE :**

import tkinter

top = tkinter.Tk()

# Code to add widgets will go here...

top.mainloop()

It will generate :



**SOURCE CODE**

**TABLE CREATED SOURCE CODE:**

import sqlite3

from tkinter import\*

from tkinter import messagebox

x=sqlite3.connect("criminals.db")

#x.execute('''create table Criminal\_Record

#(criminal\_number int primary key,name text not null,

# age int not null,

#crime text not null,fathers\_name text not null,

#city text,mobile\_number int,time\_of\_punishment text);''')

#print("Table created")

#x.close()

x.execute("insert into Criminal\_Record values(101,'abc',23,'hacker','ABD','Indore',9586265365,'6 months');")

x.execute("insert into Criminal\_Record values(102,'xyz',35,'Murderer','RBJ','Bhopal',8457922231,'10 years');")

x.execute("insert into Criminal\_Record values(103,'pqr',40,'Robber','AMI','Patna',6652348541,'1 year');")

x.execute("insert into Criminal\_Record values(104,'abm',31,'Robber','MDJ','Goa',7845965823,'6 months');")

x.execute("insert into Criminal\_Record values(105,'rst',49,'Murderer','KQR','Bhopal',8885896588,'6 years');")

x.commit()

x.close()

#cursor=x.execute("select num,name from saloni;")

#for row in cursor:

# print(row)

#x.close()

**MAIN PROGRAM SOURCE CODE:**

from tkinter import\*

import sqlite3

import tkinter.messagebox

x=sqlite3.connect('criminals.db')

c=x.cursor()

class Criminal:

def \_\_init\_\_(self,master):

self.master=master

self.left=Frame(master,width=1000,height=720,bg="lightgrey")

self.left.pack()

self.right=Frame(master,width=1000,height=720,bg="lightgrey")

self.right.pack()

self.heading=Label(self.left,text="OXFORD JAIL-CRIMINAL RECORDS",font=("arial 40 bold"),fg="black",bg="lightgrey")

self.heading.place(x=0,y=0)

self.heading=Label(self.left,text="ADD CRIMINAL RECORD",font=("arial 20 bold"),fg="blue",bg="lightgrey")

self.heading.place(x=0,y=60)

self.criminal\_number=Label(self.left,text='Criminal Number',font=("arial 18 bold"),fg="black",bg="lightgrey")

self.criminal\_number.place(x=0,y=100)

self.name=Label(self.left,text="Name",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.name.place(x=0,y=140)

self.age=Label(self.left,text="Age",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.age.place(x=0,y=180)

self.crime=Label(self.left,text="Crime",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.crime.place(x=0,y=220)

self.fathers\_name=Label(self.left,text="Father's name",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.fathers\_name.place(x=0,y=260)

self.city=Label(self.left,text="City",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.city.place(x=0,y=300)

self.mobile\_number=Label(self.left,text="Mobile Number",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.mobile\_number.place(x=0,y=340)

self.time\_of\_punishment=Label(self.left,text="Time of Punishment",font=("arial 18 bold"),fg="black",bg="lightgrey")

self.time\_of\_punishment.place(x=0,y=380)

self.cn\_1=Entry(self.left,bd=5)

self.cn\_1.place(x=250,y=100)

self.name\_1=Entry(self.left,bd=5)

self.name\_1.place(x=250,y=140)

self.age\_1=Entry(self.left,bd=5)

self.age\_1.place(x=250,y=180)

self.crime\_1=Entry(self.left,bd=5)

self.crime\_1.place(x=250,y=220)

self.fathers\_name\_1=Entry(self.left,bd=5)

self.fathers\_name\_1.place(x=250,y=260)

self.city\_1=Entry(self.left,bd=5)

self.city\_1.place(x=250,y=300)

self.mobile\_number\_1=Entry(self.left,bd=5)

self.mobile\_number\_1.place(x=250,y=340)

self.time\_of\_punishment\_1=Entry(self.left,bd=5)

self.time\_of\_punishment\_1.place(x=250,y=380)

self.submit=Button(self.left,text="Add Criminal Record",bd=5,height=2,fg="white",bg="blue",command=self.add\_criminal\_record)

self.submit.place(x=250,y=420)

self.heading1=Label(self.left,text="Click on next Link",font=("arial 20 bold"),fg="blue",bg="lightgrey")

self.heading1.place(x=150,y=480)

self.submit1=Button(self.left,text="Next Page",bd=5,height=2,fg="white",bg="blue",command=nextpage)

self.submit1.place(x=350,y=540)

def add\_criminal\_record(self):

self.value1=self.cn\_1.get()

self.value2=self.name\_1.get()

self.value3=self.age\_1.get()

self.value4=self.crime\_1.get()

self.value5=self.fathers\_name\_1.get()

self.value6=self.city\_1.get()

self.value7=self.mobile\_number\_1.get()

self.value8=self.time\_of\_punishment\_1.get()

if(self.value1=='' or self.value2==''or self.value3==''or self.value4==''or self.value5==''or self.value6==''or self.value7==''or self.value8==''):

tkinter.messagebox.showinfo("Not done","Please fill all values")

else:

sql="insert into Criminal\_Record(criminal\_number,name,age,crime,fathers\_name,city,mobile\_number,time\_of\_punishment) values(?,?,?,?,?,?,?,?)"

c.execute(sql,(self.value1,self.value2,self.value3,self.value4,self.value5,self.value6,self.value7,self.value8))

x.commit()

tkinter.messagebox.showinfo("Completed","Record has been added")

def nextpage():

class Criminal1:

def \_\_init\_\_(self,master):

self.master=master

self.left1=Frame(master,width=1000,height=720,bg="lightgrey")

self.left1.pack()

self.right1=Frame(master,width=1000,height=720,bg="lightgrey")

self.right1.pack()

self.heading=Label(self.right1,text="SEARCH RECORD",font=("arial 20 bold"),fg="blue",bg="lightgrey")

self.heading.place(x=550,y=0)

self.name=Label(master,text="Enter Criminal Name",font=("arial 18 bold"),bg="lightgrey")

self.name.place(x=550,y=100)

self.namenet=Entry(master,bd=5)

self.namenet.place(x=850,y=100)

self.search=Button(master,text="Search Criminal Record",bd=5,height=2,fg="white",bg="blue",command=self.search\_criminal\_record)

self.search.place(x=750,y=160)

def search\_criminal\_record(self):

self.input=self.namenet.get()

if(self.input==''):

tkinter.messagebox.showinfo("Not done","Please enter the name")

else:

sql="SELECT\*FROM Criminal\_Record WHERE name LIKE ?"

self.res=c.execute(sql,(self.input,))

for self.row in self.res:

self.criminal\_number=self.row[0]

self.name=self.row[1]

self.age=self.row[2]

self.crime=self.row[3]

self.fathers\_name=self.row[4]

self.city=self.row[5]

self.mobile\_number=self.row[6]

self.time\_of\_punishment=self.row[7]

self.u\_criminal\_number=Label(self.master,text="Criminal Number",font=("arial 18 bold"),bg="lightgrey")

self.u\_criminal\_number.place(x=550,y=220)

self.u\_name=Label(self.master,text="Name",font=("arial 18 bold"),bg="lightgrey")

self.u\_name.place(x=550,y=260)

self.u\_age=Label(self.master,text="Age",font=("arial 18 bold"),bg="lightgrey")

self.u\_age.place(x=550,y=300)

self.u\_crime=Label(self.master,text="Crime",font=("arial 18 bold"),bg="lightgrey")

self.u\_crime.place(x=550,y=340)

self.u\_fathers\_name=Label(self.master,text="Father's Name",font=("arial 18 bold"),bg="lightgrey")

self.u\_fathers\_name.place(x=550,y=380)

self.u\_city=Label(self.master,text="City",font=("arial 18 bold"),bg="lightgrey")

self.u\_city.place(x=550,y=420)

self.u\_mobile\_number=Label(self.master,text="Mobile Number",font=("arial 18 bold"),bg="lightgrey")

self.u\_mobile\_number.place(x=550,y=460)

self.u\_time\_of\_punishment=Label(self.master,text="Time of punishment",font=("arial 18 bold"),bg="lightgrey")

self.u\_time\_of\_punishment.place(x=550,y=500)

self.ent1=Entry(self.master,bd=5)

self.ent1.place(x=850,y=220)

self.ent1.insert(END,str(self.criminal\_number))

self.ent2=Entry(self.master,bd=5)

self.ent2.place(x=850,y=260)

self.ent2.insert(END,str(self.name))

self.ent3=Entry(self.master,bd=5)

self.ent3.place(x=850,y=300)

self.ent3.insert(END,str(self.age))

self.ent4=Entry(self.master,bd=5)

self.ent4.place(x=850,y=340)

self.ent4.insert(END,str(self.crime))

self.ent5=Entry(self.master,bd=5)

self.ent5.place(x=850,y=380)

self.ent5.insert(END,str(self.fathers\_name))

self.ent6=Entry(self.master,bd=5)

self.ent6.place(x=850,y=420)

self.ent6.insert(END,str(self.city))

self.ent7=Entry(self.master,bd=5)

self.ent7.place(x=850,y=460)

self.ent7.insert(END,str(self.mobile\_number))

self.ent8=Entry(self.master,bd=5)

self.ent8.place(x=850,y=500)

self.ent8.insert(END,str(self.time\_of\_punishment))

self.update=Button(self.master,text="UPDATE RECORD" ,bd=5,height=2,fg="white",bg="blue",command=self.update\_criminal\_record)

self.update.place(x=650,y=580)

self.update=Button(self.master,text="DELETE RECORD" ,bd=5,height=2,fg="white",bg="red",command=self.delete\_criminal\_record)

self.update.place(x=920,y=580)

def update\_criminal\_record(self):

self.variable1=self.ent1.get()

self.variable2=self.ent2.get()

self.variable3=self.ent3.get()

self.variable4=self.ent4.get()

self.variable5=self.ent5.get()

self.variable6=self.ent6.get()

self.variable7=self.ent7.get()

self.variable8=self.ent8.get()

sql2="UPDATE Criminal\_Record SET criminal\_number=?,name=?,age=?,crime=?,fathers\_name=?,city=?,mobile\_number=?,time\_of\_punishment=? WHERE name LIKE ?"

c.execute(sql2,(self.variable1,self.variable2,self.variable3,self.variable4,self.variable5,self.variable6,self.variable7,self.variable8,self.namenet.get()))

x.commit()

tkinter.messagebox.showinfo("Updated","Successfully updated the values")

def delete\_criminal\_record(self):

sql3="DELETE from Criminal\_Record WHERE name LIKE ?"

c.execute(sql3,(self.namenet.get(),))

x.commit()

tkinter.messagebox.showinfo("Deleted","Record is deleted")

self.ent1.destroy()

self.ent2.destroy()

self.ent3.destroy()

self.ent4.destroy()

self.ent5.destroy()

self.ent6.destroy()

self.ent7.destroy()

self.ent8.destroy()

obj2=Criminal1(root)

root.geometry("1200x900")

root=Tk()

obj1=Criminal(root)

root.geometry("1200x900”)

**USED SYNTAX EXPLANATION**

We have created one database as criminals and we connect it to sqlite3 with connect keyword and “criminals.db”, So it can simply connect the database.

We have used classes as Criminal and Criminal1, Criminal class can add the records of criminals by using self keyword we connect the GUI as well.

Criminal1 class can delete, search and update the data of criminals.

We have used one function nextpage() which declares the class Criminal1 so that the link of next button will be the explanation of Criminal1 class.

We have also used the message-box so that the message has been shown that either it may be added, deleted or updated.

We used different SQL (structured query language ) queries to define the database table modifications.

The table will be generated on the other slide which can be open using SQlite3 browser 3.6.0 (This version we have used).

**RESULT**

The result of this project is that the GUI and the code will work together and give us the required project.

The first source code is of the table created in database.

The second source code is of the full program with GUI .

**BIBLIOGRAPHY**

**Books:**

* Introduction to Programming using Python by Y. Daniel Liang, Pearson.
* Problem solving and Python Programming by Ashok Kamthane and Amit Kamthane, Tata McGraw Hill, India.

**Websites:**

* Youtube : <http://www.youtube.com/>
* Tutorial’s point : <http://www.tutorialspoint.com/python/python_gui_programming.htm>
* Sanfoundary : http://www.sanfoundry.com/python-problems-solutions/