Term Id: 220211 **Subject:** INT213

CA-1

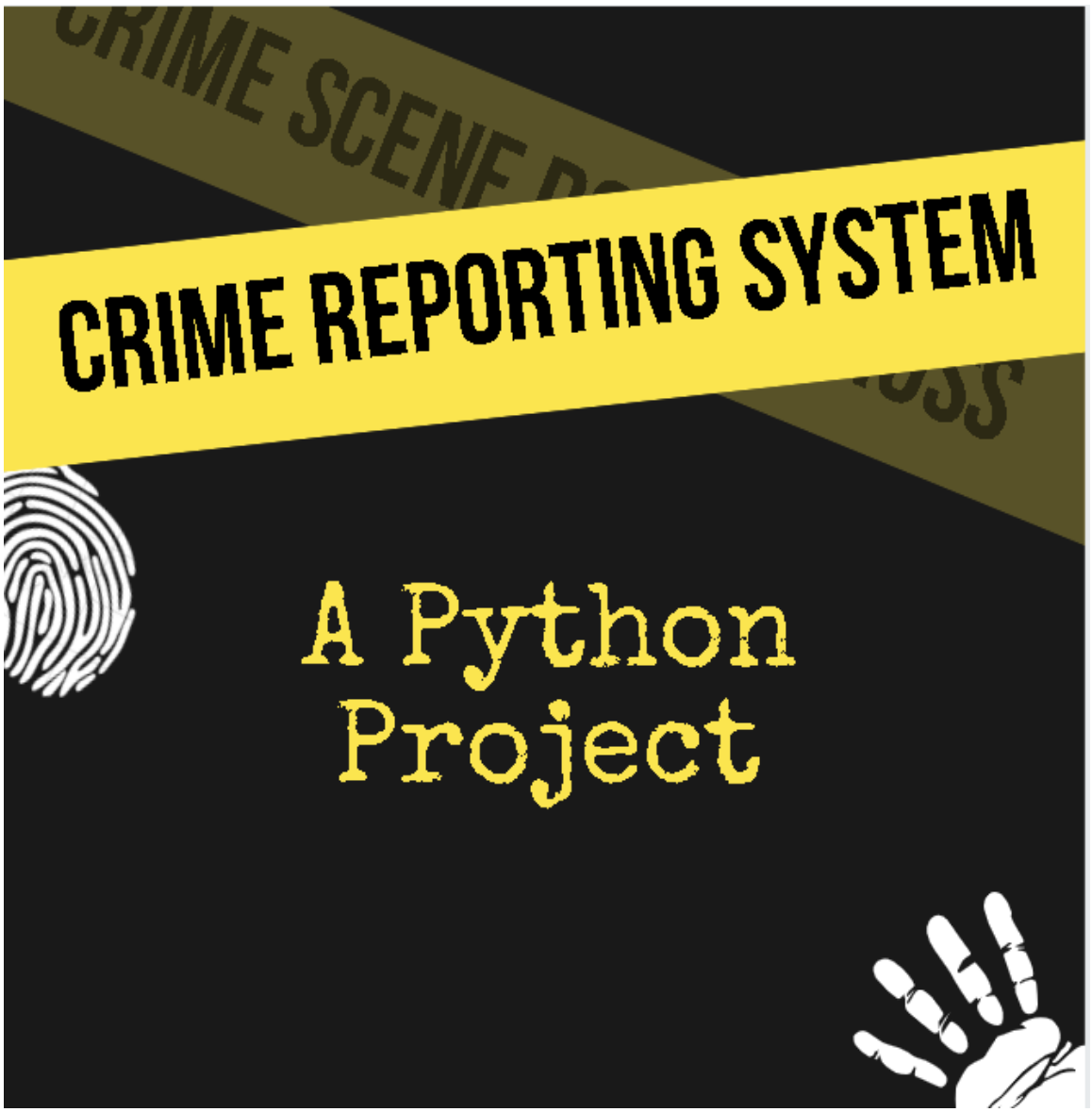
Group Members:

Name	Roll No	Reg. No.
Alan James	20	11903972
Joy Garg	62	11918595

Project Name: Crime Reporting System

Section: K19KH

Submitted to: Mrs. Ankita Wadhawan



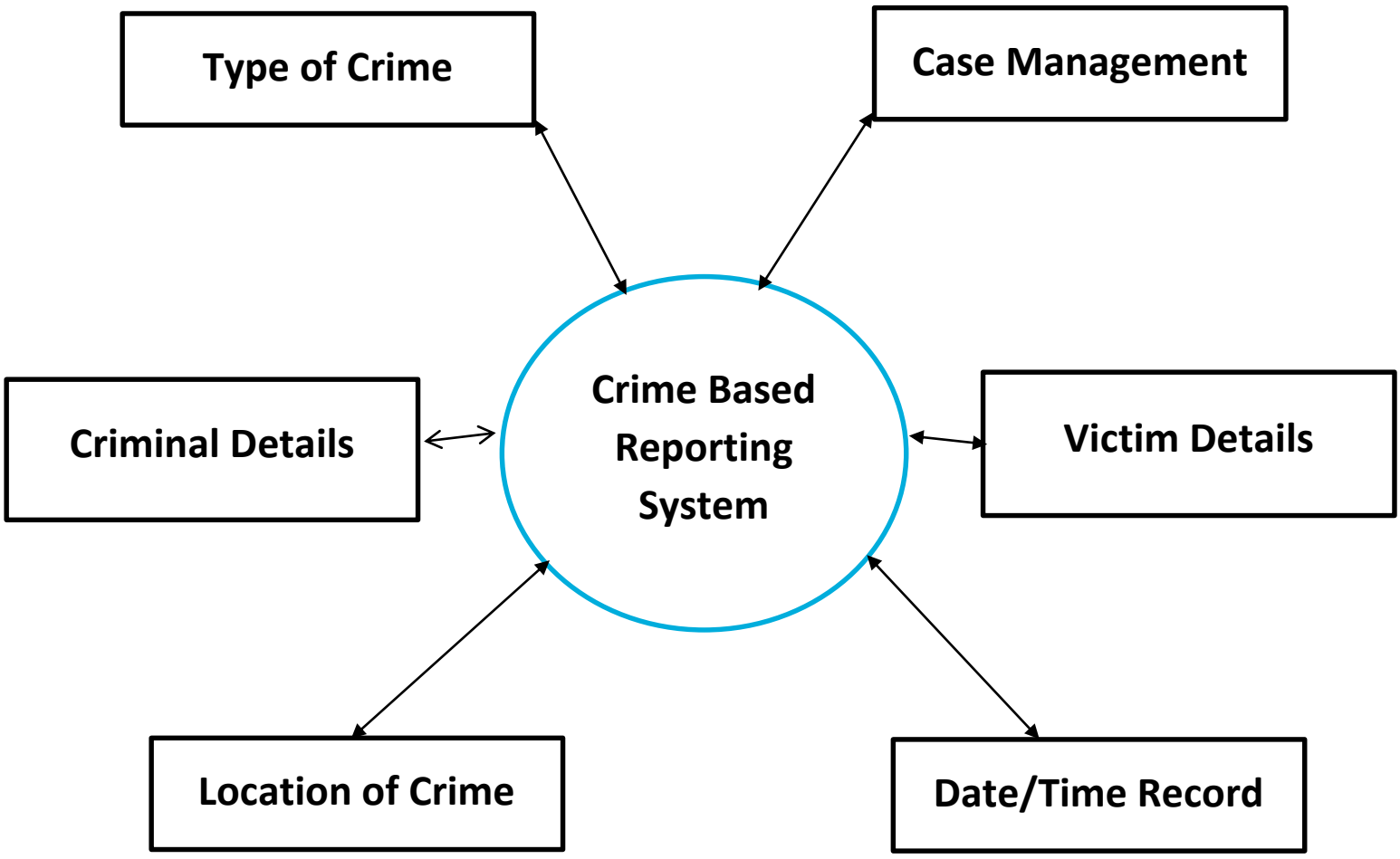
CRIME REPORTING SYSTEM

Built using Python with the additional libraries tkinter and sqlite.

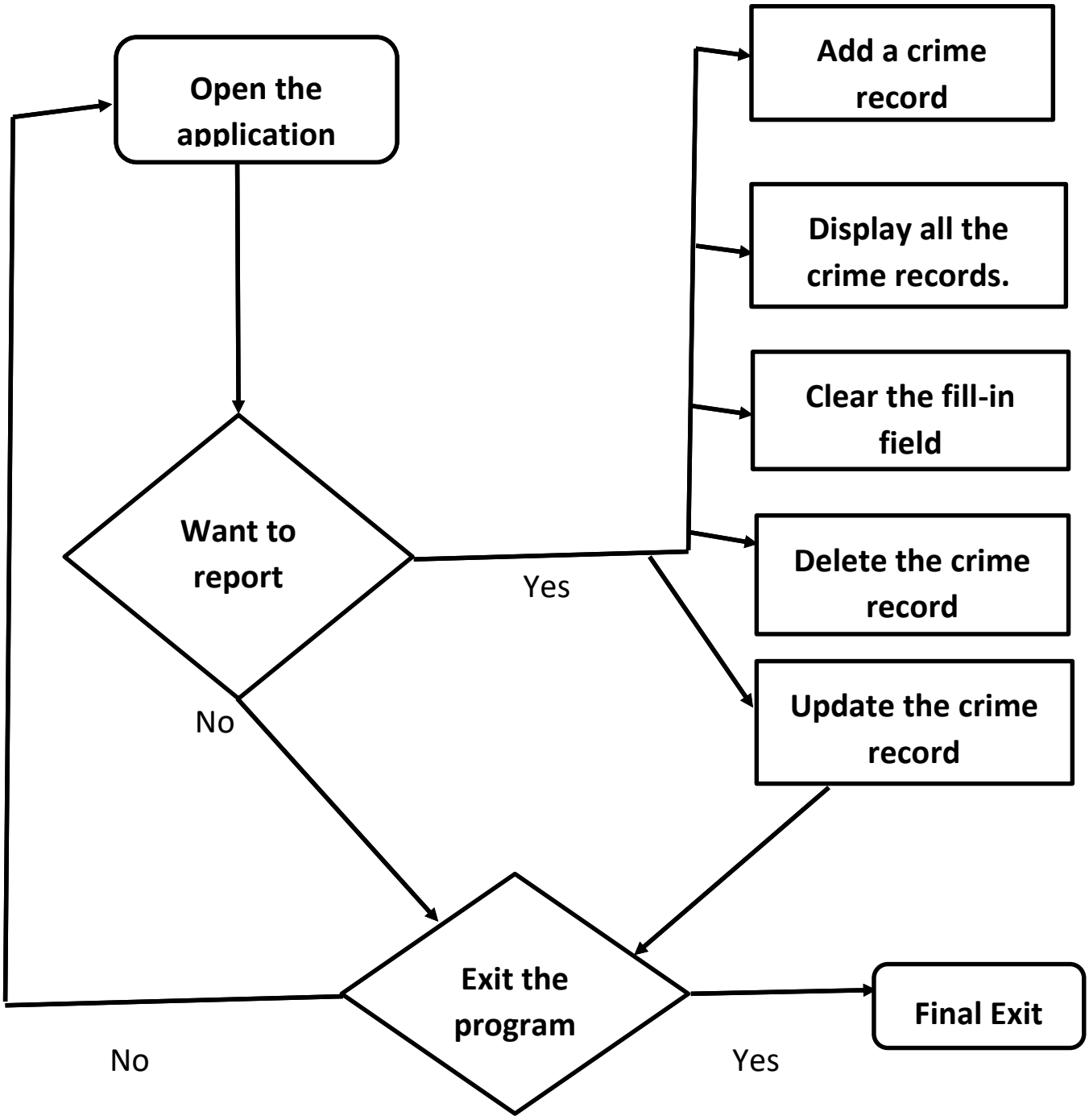
INTRODUCTION

Crime is an unlawful and punishable offence under law, and the state. Crime Reporting is an actual respective reporting of the incident for better investigation, prosecution and judgement. Over the years, crime rate has significantly grown to stooping levels. The Crime Reporting System discussed here is a simple approach for any basic level application to complex. It helps you to carefully fill in the first investigation process findings, with details. This helps the police forces to capture, identify the criminal or person behind the crime. This Python project also uses database to its advantage to store the details permanently and do the necessary updating or clarification later on.

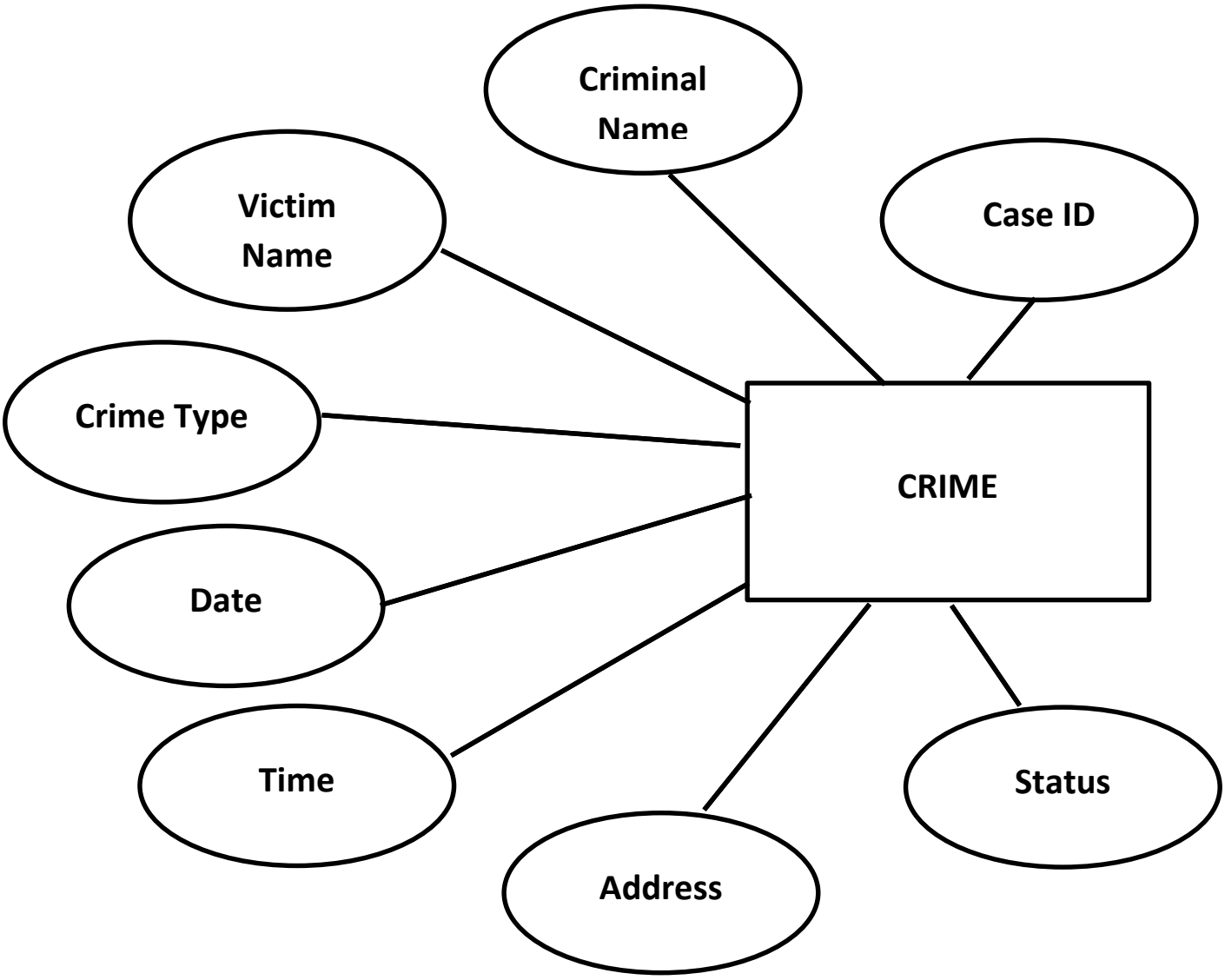
DATA FLOW DIAGRAM



FLOW CHART


Open the application
Add a crime record
Display all the crime records.
Clear the fill-in field
Delete the crime record
Update the crime record
Want to report
Yes
No
Exit the program
No
Yes
Final Exit

ENTITY-RELATIONSHIP DIAGRAM



SCREENSHOTS

Programming Code:

Frontend

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StudentsDatabaseManagementSystems.py > Student > __init__

```
from tkinter import*
import tkinter.messagebox
import StdDatabase
#Frontend


class Student:

    def __init__(self,root):
        self.root =root
        self.root.title("Crime Reporting System")
        self.root.geometry("1350x750+0+0")
        self.root.config(bg="maroon")

        StdID = StringVar()
        Firstname = StringVar()
        Surname = StringVar()
        DoB = StringVar()
        Age = StringVar()
        Gender = StringVar()
        Address = StringVar()
        Mobile = StringVar()

        #=======================================Function Declaration=============================

        def iExit():
            iExit = tkinter.messagebox.askyesno("Crime Reporting Sytems","Confirm if you want to exit")
            if iExit > 0:
                root.destroy()
                return

        def ClearData():
            self.txtStdID.delete(0,END)
            self.txtfna.delete(0,END)
            self.txtsna.delete(0,END)
            self.txtDoB.delete(0,END)
            self.txtAge.delete(0,END)
            self.txtGender.delete(0,END)
```

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StudentsDatabaseManagementSystems.py > Student > __init__

```
            self.txtAdr.delete(0,END)
            self.txtMobile.delete(0,END)

        def addData():
            if(len(StdID.get())!=0):
                StdDatabase.addStdRec(StdID.get(), Firstname.get(),  Surname.get(), DoB.get(), Age.get(), Gender.get(), Address.get(), Mobile.get())
                studentlist.delete(0,END)
                studentlist.insert(END,(StdID.get(), Firstname.get(),  Surname.get(), DoB.get(), Age.get(), Gender.get(), Address.get(), Mobile.get()))

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

        def StudentRec(event):
            global sd
            searchStd = studentlist.curselection()[0]
            sd = studentlist.get(searchStd )

            self.txtStdID.delete(0,END)
            self.txtStdID.insert(END,sd[1])
            self.txtfna.delete(0,END)
            self.txtfna.insert(END,sd[2])
            self.txtsna.delete(0,END)
            self.txtsna.insert(END,sd[3])
            self.txtDoB.delete(0,END)
            self.txtDoB.insert(END,sd[4])
            self.txtAge.delete(0,END)
            self.txtAge.insert(END,sd[5])
            self.txtGender.delete(0,END)
            self.txtGender.insert(END,sd[6])
            self.txtAdr.delete(0,END)
            self.txtAdr.insert(END,sd[7])
            self.txtMobile.delete(0,END)
            self.txtMobile.insert(END,sd[8])
```

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StudentsDatabaseManagementSystems.py > Student > __init__

```
75          def DeleteDate():
76              if(len(StdID.get())!=0):
77                  StdDatabase.deleteRec(sd[0])
78                  ClearData()
79                  DisplayData()
80
81          def searchDatabase():
82              studentlist.delete(0,END)
83              for row in StdDatabase.searchData(StdID.get(), Firstname.get(),Surname.get(),DoB.get(), Age.get(), Gender.get(), Address.get(), Mobile.get()):
84                  studentlist.insert(END,row,str(""))
85
86          #def update():
87              #if(len(StdID.get())!=0):
88                #StdDatabase.dataUpdate(sd[0],(StdID.get(), Firstname.get(),Surname.get(),DoB.get(), Age.get(),Gender.get(),Address.get(),Mobile.get()))
89
90          def update():
91              if(len(StdID.get())!=0):
92                  StdDatabase.deleteRec(sd[0])
93              if(len(StdID.get())!=0):
94                  StdDatabase.addStdRec(StdID.get(), Firstname.get(),  Surname.get(), DoB.get(), Age.get(), Gender.get(), Address.get(), Mobile.get())
95                  studentlist.delete(0,END)
96                  studentlist.insert(END,(StdID.get(), Firstname.get(),  Surname.get(), DoB.get(), Age.get(), Gender.get(), Address.get(), Mobile.get()))
97
98
99          #=====================================Frame=============================================================
100         MainFrame =Frame(self.root, bg="maroon")
101         MainFrame.grid()
102
103         TitFrame = Frame(MainFrame, bd=2,  padx=54,pady=8, bg="Ghost White", relief=RIDGE)
104         TitFrame.pack(side=TOP)
105
106         self.lblTit =Label(TitFrame,font=('arial', 47,'bold'),text="Crime Reporting System",bg="Ghost White")
107         self.lblTit.grid(sticky=W)
108
109         ButtonFrame =Frame(MainFrame, bd=2, width=1350, height=70, padx=18,pady=10, bg="Ghost White", relief=RIDGE)
110         ButtonFrame.pack(side=BOTTOM)
```

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StudentsDatabaseManagementSystems.py > Student > __init__

```
112         DataFrame =Frame(MainFrame, bd=1, width=1300, height=400, padx=20, pady=20, relief=RIDGE, bg="maroon")
113         DataFrame.pack(side=BOTTOM)
114
115         DataFrameLEFT =LabelFrame(DataFrame, bd=1, width=1000, height=600, padx=20, relief=RIDGE
116                                   , font=('arial', 20,'bold'), text="Crime Info:\n",bg="Ghost White")
117         DataFrameLEFT.pack(side=LEFT)
118
119         DataFrameRIGHT =LabelFrame(DataFrame, bd=1, width=450, height=300, padx=31,pady=3, relief=RIDGE
120                                    , font=('arial', 20,'bold'),bg="Ghost White", text="Crime Details:\n",)
121         DataFrameRIGHT.pack(side=RIGHT)
122         #=====================================Labels and Entrys=============================================
123         self.lblStdID =Label(DataFrameLEFT,font=('arial',20,'bold'),text="Case ID:",padx=2,pady=2,bg="Ghost White")
124         self.lblStdID.grid(row=0, column =0,sticky=W)
125         self.txtStdID =Entry(DataFrameLEFT,font=('arial',20,'bold'), textvariable=StdID, width=39)
126         self.txtStdID.grid(row=0, column =1)
127
128         self.lblfna =Label(DataFrameLEFT, font=('arial', 20,'bold'), text="Name:",padx=2,pady=2 ,bg="Ghost White")
129         self.lblfna.grid(row=1, column=0,sticky=W)
130         self.txtfna=Entry(DataFrameLEFT, font=('arial', 20,'bold'),textvariable =Firstname , width=39)
131         self.txtfna.grid(row=1, column=1)
132
133         self.lblsna =Label(DataFrameLEFT, font=('arial', 20,'bold'), text="Type of Crime:",padx=2,pady=2 ,bg="Ghost White")
134         self.lblsna.grid(row=2, column=0,sticky=W)
135         self.txtsna=Entry(DataFrameLEFT, font=('arial', 20,'bold'),textvariable = Surname, width=39)
136         self.txtsna.grid(row=2, column=1)
137
138         self.lblDoB =Label(DataFrameLEFT, font=('arial', 20,'bold'),text="Date of Crime:",padx=2,pady=3 ,bg="Ghost White")
139         self.lblDoB.grid(row=3, column=0,sticky=W)
140         self.txtDoB=Entry(DataFrameLEFT,font=('arial', 20,'bold'),textvariable = DoB,width=39)
141         self.txtDoB.grid(row=3, column=1)
142
143         self.lblAge =Label(DataFrameLEFT, font=('arial', 20,'bold'), text="Time:",padx=2,pady=3 ,bg="Ghost White")
144         self.lblAge.grid(row=4, column=0,sticky=W)
145         self.txtAge=Entry(DataFrameLEFT, font=('arial', 20,'bold'),textvariable = Age, width=39)
146         self.txtAge.grid(row=4, column=1)
```

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StudentsDatabaseManagementSystems.py > Student > __init__

```
148         self.lblGender =Label(DataFrameLEFT, font=('arial', 20,'bold'),text="VictimsName:",padx=2,pady=3 ,bg="Ghost White")
149         self.lblGender.grid(row=5, column=0,sticky=W)
150         self.txtGender=Entry(DataFrameLEFT, font=('arial', 20,'bold'),textvariable = Gender, width=39)
151         self.txtGender.grid(row=5, column=1)
152 
153         self.lblAdr =Label(DataFrameLEFT, font=('arial', 20,'bold'), text="Address:",padx=2,pady=3 ,bg="Ghost White")
154         self.lblAdr.grid(row=6, column=0,sticky=W)
155         self.txtAdr=Entry(DataFrameLEFT, font=('arial', 20,'bold'),textvariable = Address, width=39)
156         self.txtAdr.grid(row=6, column=1)
157 
158         self.lblMobile =Label(DataFrameLEFT, font=('arial', 20,'bold'),text="Status :",padx=2,pady=3 ,bg="Ghost White")
159         self.lblMobile .grid(row=7, column=0,sticky=W)
160         self.txtMobile =Entry(DataFrameLEFT, font=('arial', 20,'bold'),textvariable = Mobile , width=39)
161         self.txtMobile .grid(row=7, column=1)
162         #=======================================Listbox and Scrollbar=============================================
163 
164         scrollbar = Scrollbar(DataFrameRIGHT)
165         scrollbar.grid(row=0, column=1, sticky ='ns')
166 
167         studentlist = Listbox(DataFrameRIGHT, width = 41, height=16, font=('arial', 12,'bold'), yscrollcommand=scrollbar.set)
168         studentlist.bind('<<ListboxSelect>>', StudentRec)
169         studentlist.grid(row=0, column=0, padx=8)
170         scrollbar.config(command=studentlist.yview)
171         #=======================================Buttons Widget=====================================================
172         self.btnAddData=Button(ButtonFrame, text='Add New', font=('arial', 20,'bold'),height=1, width=10, bd=4,
173                                command=addData)
174         self.btnAddData.grid(row=0,column=0)
175 
176         self.btnDisplayData=Button(ButtonFrame, text='Display', font=('arial', 20,'bold'),height=1, width=10,
177                                    bd=4, command=DisplayData)
178         self.btnDisplayData.grid(row=0,column=1)
179 
180         self.btnClearData=Button(ButtonFrame, text='Clear', font=('arial', 20,'bold'),height=1, width=10, bd=4,
181                                  command=ClearData)
182         self.btnClearData.grid(row=0,column=2)
```

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StudentsDatabaseManagementSystems.py > Student > __init__

```
184         self.btnDeleteData=Button(ButtonFrame, text='Delete', font=('arial', 20,'bold'),height=1, width=10, bd=4,
185                                   command=DeleteDate)
186         self.btnDeleteData.grid(row=0,column=3)
187 
188         self.btnSearchData=Button(ButtonFrame, text='Search', font=('arial', 20,'bold'),height=1, width=10, bd=4,
189                                   command =searchDatabase)
190         self.btnSearchData.grid(row=0,column=4)
191 
192         self.btnUpdateData=Button(ButtonFrame, text='Update', font=('arial', 20,'bold'),height=1, width=10, bd=4,
193                                   command = update)
194         self.btnUpdateData.grid(row=0,column=5)
195 
196         self.btnExit=Button(ButtonFrame, text='Exit', font=('arial', 20,'bold'),height=1, width=10, bd=4, command=iExit)
197         self.btnExit.grid(row=0,column=6)
198 
199 
200 
201 
202 if __name__=='__main__':
203     root = Tk()
204     application = Student(root)
205     root.mainloop()
```

Backend

StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StdDatabase.py > ...

```python
import sqlite3
#backend

def studentData():
    con=sqlite3.connect("student.db")
    cur = con.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS student (id INTEGER PRIMARY KEY, StdID text,Firstname text,Surname text,DoB text, \
        Age text,Gender text,Address text,Mobile text)")
    con.commit()
    con.close()

def addStdRec(StdID, Firstname,Surname,DoB, Age,Gender, Address, Mobile):
    con=sqlite3.connect("student.db")
    cur = con.cursor()
    cur.execute("INSERT INTO student VALUES (NULL,?,?,?,?,?,?,?,?)", \
                (StdID, Firstname,Surname,DoB, Age,Gender, Address, Mobile))
    con.commit()
    con.close()

def viewData():
    con=sqlite3.connect("student.db")
    cur = con.cursor()
    cur.execute("SELECT * FROM student")
    rows=cur.fetchall()
    con.close
    return rows

def deleteRec(id):
    con=sqlite3.connect("student.db")
    cur = con.cursor()
    cur.execute("DELETE FROM student WHERE id=?", (id,))
    con.commit()
    con.close


def searchData(StdID="", Firstname="",Surname="",DoB="", Age="",Gender="", Address="", Mobile=""):
    con=sqlite3.connect("student.db")
    cur = con.cursor()
```

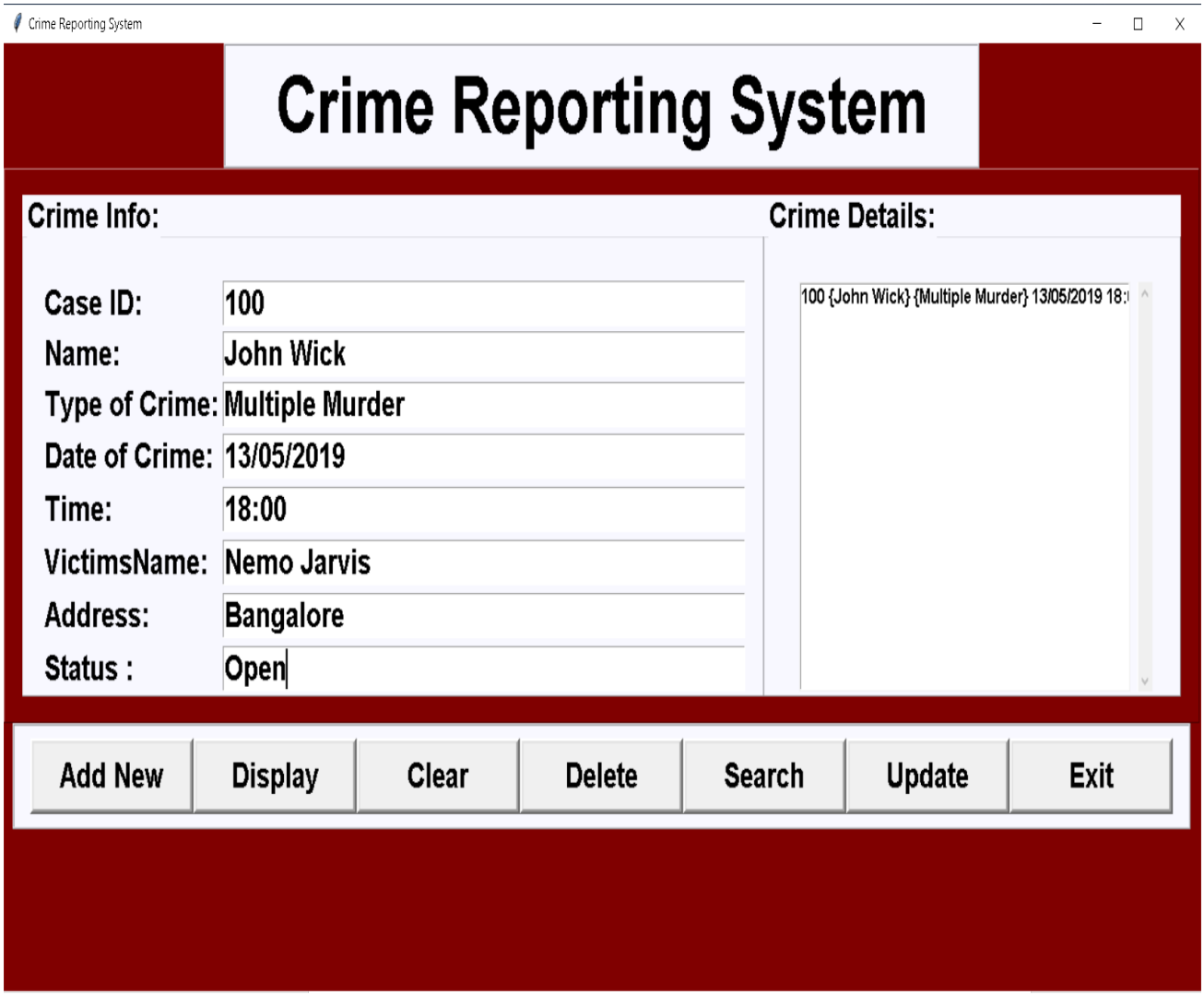
StudentsDatabaseManagementSystems.py | StdDatabase.py

C: > Users > alaan > Desktop > INT208 Project > StdDatabase.py > ...

```python
    cur.execute("SELECT * FROM student WHERE StdID=? OR Firstname=? OR Surname=? OR DoB=? OR Age=? OR Gender=? OR Address=? OR Mobile=? ", \
                (StdID, Firstname,Surname,DoB, Age,Gender, Address, Mobile))
    rows=cur.fetchall()
    con.close()
    return rows

def dataUpdate(id,StdID="", Firstname="",Surname="",DoB="", Age="",Gender="", Address="", Mobile=""):
    con=sqlite3.connect("student.db")
    cur = con.cursor()
    cur.execute("UPDATE student SET StdID=?, Firstname=?,Surname=?,DoB=?, Age=?,Gender=?, Address=?, Mobile=?, WHERE id=?", \
                (StdID, Firstname,Surname,DoB, Age,Gender, Address, Mobile, id))
    con.commit()
    con.close()




studentData()
```

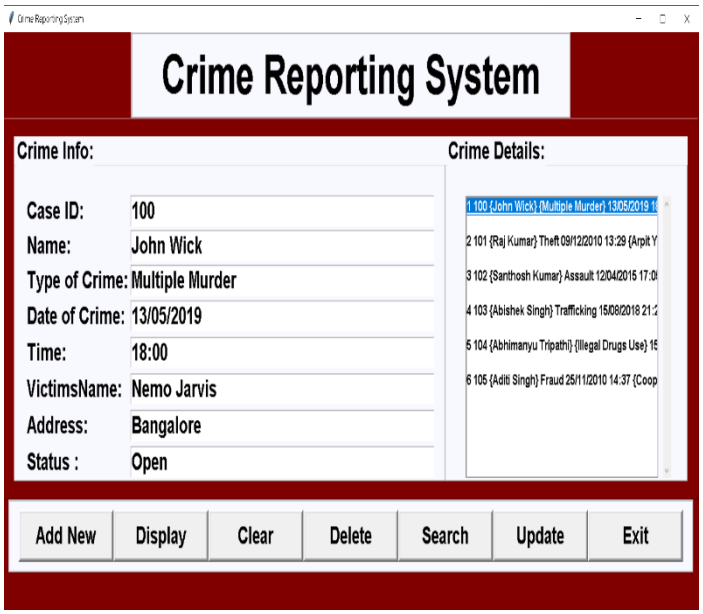
Output:

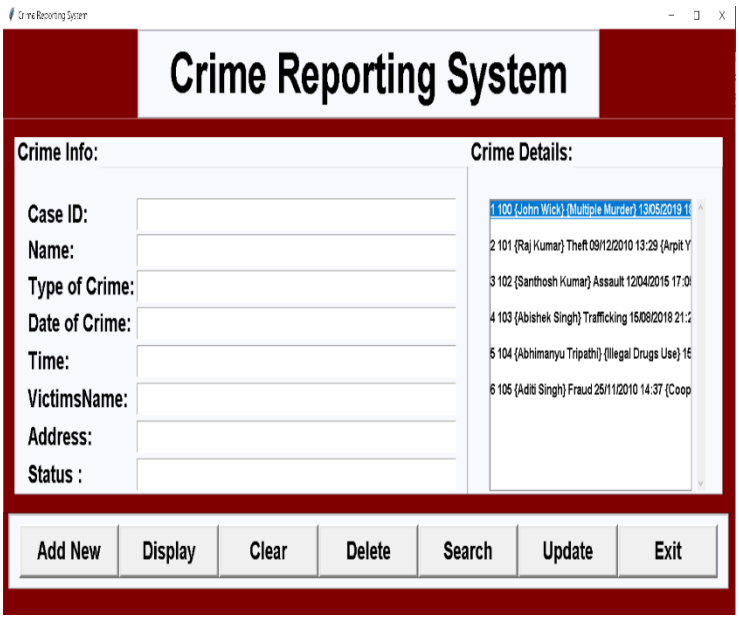
- Adding some records to the database. After clicking the **add** button, the record gets stored onto the database. This is indicated by the display of record in the Crime Details Space.



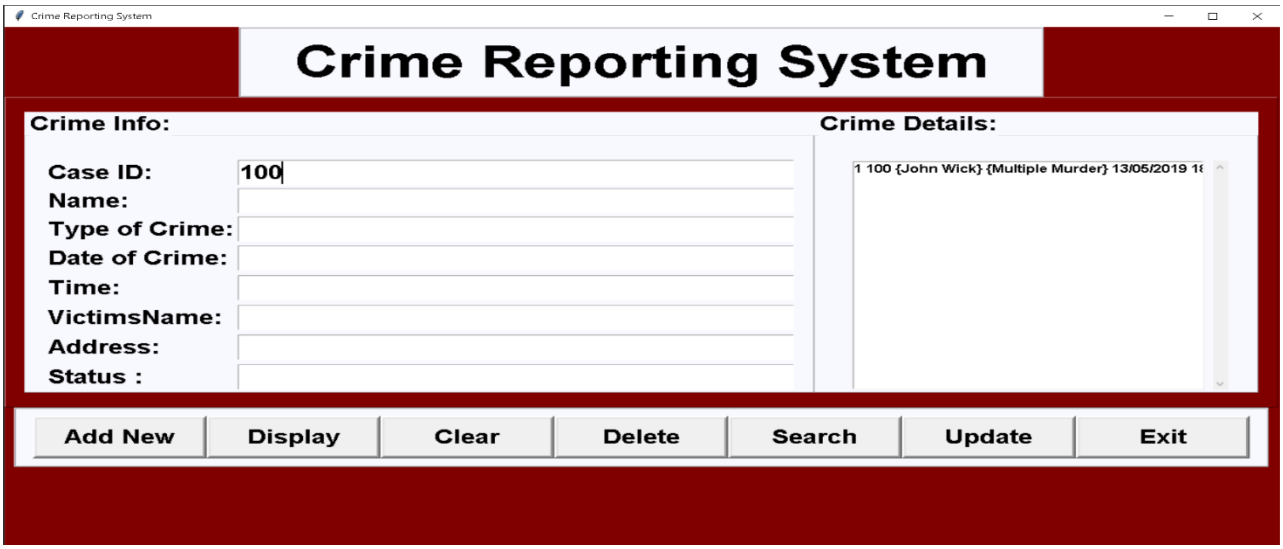
- After adding some records, when **Display** button is clicked. We retrieve all the data from database and display it at the Crime Details space on the right.

- When we click the **clear** button, the added entry automatically clears from every field.

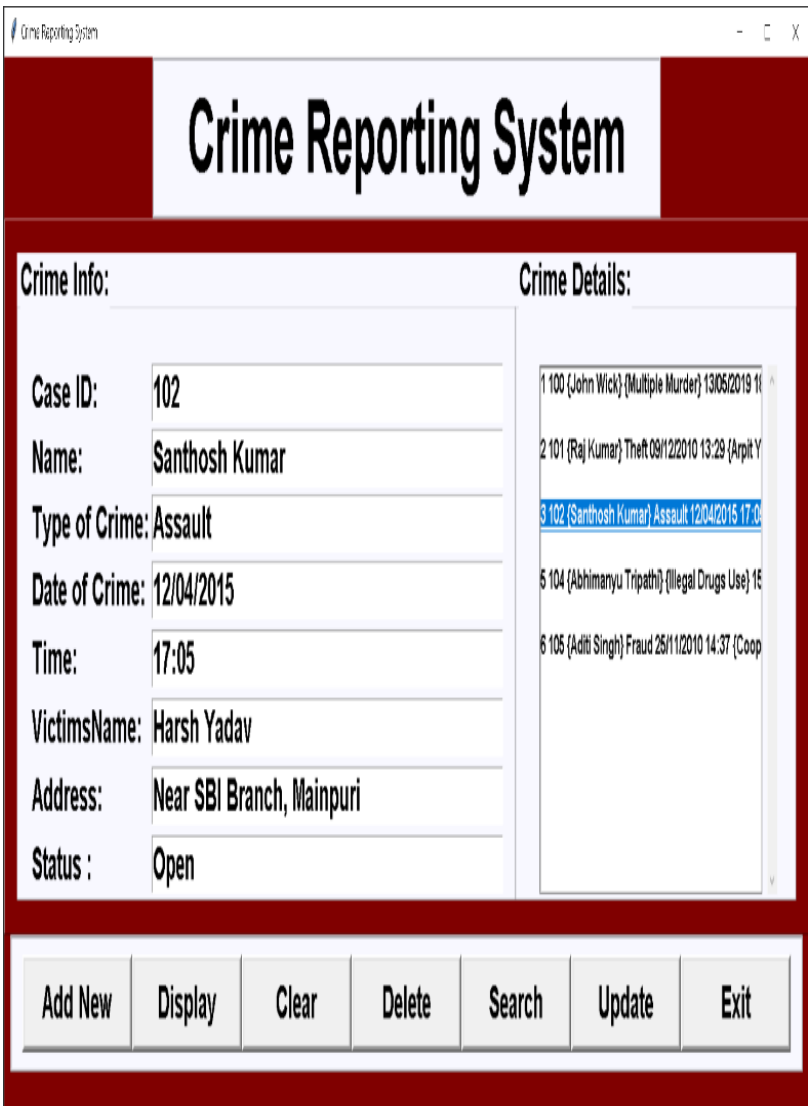


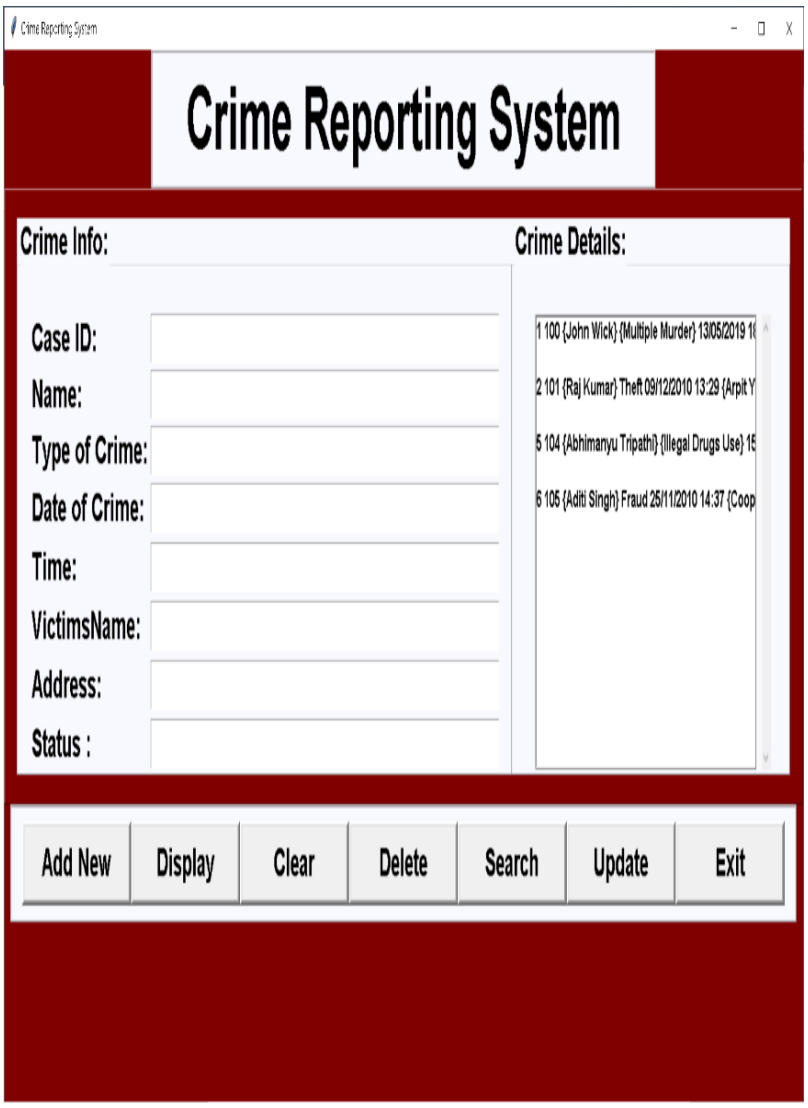


- By giving in the variable, we **search** for record shown in Crime Details space.

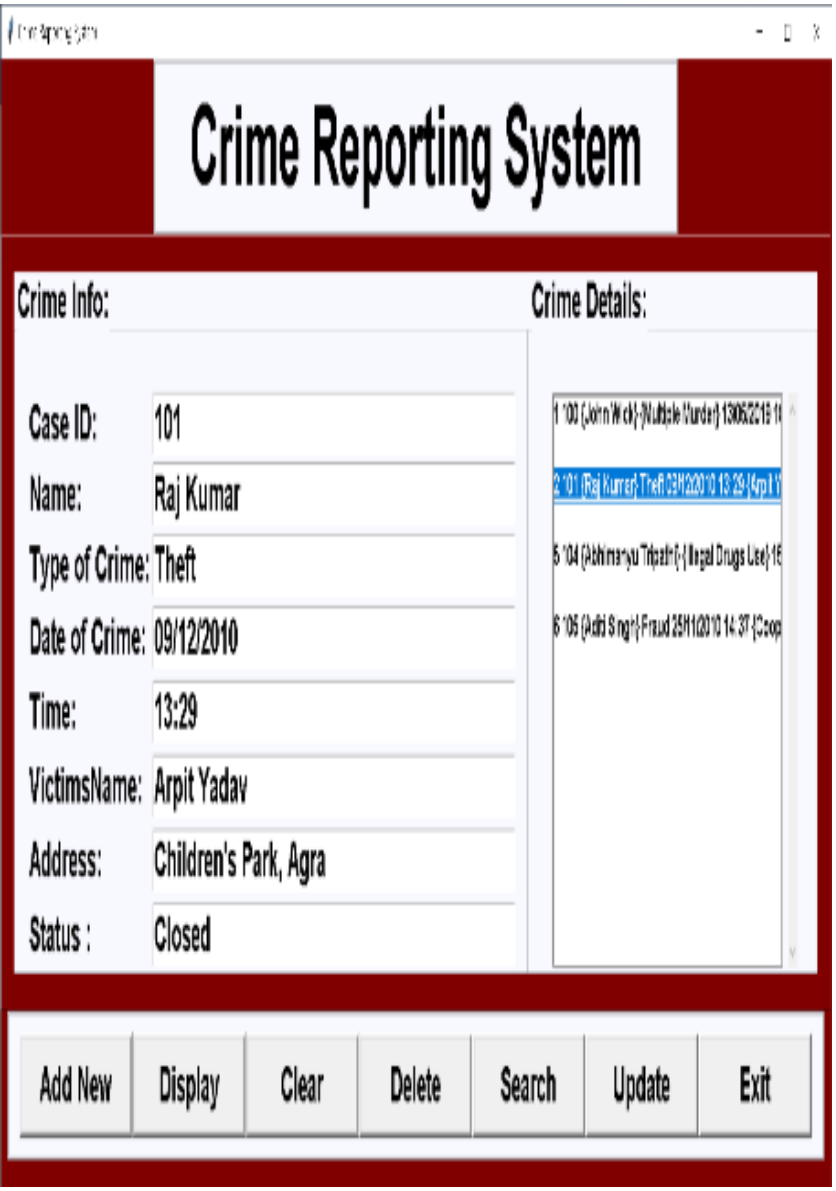


- When we click the **delete** button, that particular record is deleted from database as well as it vanishes from the Crime Details Space.





- We select the record to be updated, and click **Update** button to make changes.



- When we select the delete button, a confirmation opens up asking Yes/No. Depending on your decision the program proceeds.

RESULTS

After preparing this basic Crime Reporting System, one is able to combine the details accurately and is user-friendly. Crime Reporting depends on the victim's mind and style. It is a necessary thing to address since we all require a decent crime-free neighborhood. Through this project we also came about the different challenges one might face, when reporting a crime. The essential process being reporting, and management is possible through flexible supervision and agreement. This calls for many opportunities by integrating this project with API calls to make it more interactive and easiness for the user. Also, we can use some ML models and algorithms to better study from the existing database, and make decisions and predictions to better configure and define the security and crime-free environment in our lives.