STUDENT DETAIL RECORD

A PROJECT REPORT

Submitted by

CALAMBUR AJAY   [Reg No: RA1811003010259]
SAARTHAK MEHTA [Reg No: RA1811003010271]

Under the guidance of

Dr. SenthilKumar

(Assistant Professor, Department of Computer Science & Engineering)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2021

SRM Institute of Science and Technology

(Deemed to be University u/s 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "STUDENT DETAIL RECORD " is the bonafide work of "CALAMBUR AJAY [RA1811003010259], SAARTHAK MEHTA[RA1811003010271]"who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE   SIGNATURE

Dr. SenthilKumar   Dr. B. AMUTHA
GUIDE   HEAD OF THE DEPARTMENT
                                                                 Dept. of Computer Science & Engi-
Dept. of Computer Science & Engi-   neering
neering

Signature of the Internal Examiner   Signature of the External Examiner

ABSTRACT

In this day and age, to store multiple rows and columns of names, registration number and further required identification values, which can be used for reference in the future, get lost when not managed properly. A reference can go a long way in getting the details for a candidate and finding out about the past of this person and thus we need a database that can store these values and help in maintaining and cleaning the data properly. A student record system is required for keeping the documents of each and every alumni, past or present.

In this project, we use python and the GUI library Tkinter to make an application that can help us store as many records as possible. These will contain quite a few parameters which include registration number,name,department,gender and age. We use sqlite for the backend requirement of this project.

Introduction

SQLite3

SQLite3 can be integrated with Python using sqlite3 module, which was written by Gerhard Haring. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards.

To use sqlite3 module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements.

For all our sql application we use Sqlite3 library in python which helps with the interconnectivity of the database and helps us in the operation component.

Tkinter

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.

METHOD

We made a student database management system to:

1. Insert , new values
2. Update, present values
3. Delete , unwanted values

The data gets saved in a sql file for easier use later

We need to create a user interface. Creating a GUI application using Tkinter is an easy task. All we 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.

We use tkinter as our front end interface with buttons and customizable input and different actions namely update,rename, insert

CODE SNIPPET

```python
from tkinter import *
from tkinter.ttk import *
from sqlite3 import *
a=Tk()
a.title('tk')
reg=Label(a,text="REGNO")
regtxt=Entry(a,width=10)
reg.grid(column=0,row=0)
regtxt.grid(column=1,row=0)

name=Label(a,text="NAme")
nametxt=Entry(a,width=10)
name.grid(column=0,row=1)
nametxt.grid(column=1,row=1)

dept=Label(a,text="Dept")
depttxt=Entry(a,width=10)
dept.grid(column=0,row=2)
depttxt.grid(column=1,row=2)

gen=Label(a,text="Gender")
gen.grid(column=0,row=3)
i=IntVar()
ra1=Radiobutton(a,text="MALE",value=1,variable=i)
ra2=Radiobutton(a,text="FEMALE",value=2,variable=i)
se=i.get()
gen=''
```

```python
def c():
    se=i.get()

    if(se==1):
        gen='MALE'
    else:
        gen='FEMALE'
    return gen
ra1.grid(column=1,row=3)
ra2.grid(column=2,row=3)

age=Label(a,text="age")
w = Spinbox(a, from_=10, to=100)

age.grid(column=0,row=4)
w.grid(column=1,row=4)

con = connect('myTable.db')
cur = con.cursor()
try:
    cur.execute("""CREATE TABLE student(
    reg INTEGER,
    name VARCHAR(20),
    dept VARCHAR(20),
    gender VARCHAR(20),
    age INTEGER);""")
except OperationalError:
    None
```

```python
def insert():
    a=c()
    cur.execute("""INSERT INTO student(reg,name,dept,gender,age) VALUES(?,?,?,?,?);""",
        (regtxt.get(),nametxt.get(),depttxt.get(),a,w.get()))
```

Function defined to insert new values to the current value

```
b1=Button(a,text='Insert',command=insert)
def update():
    a=c()
    cur.execute("""UPDATE student SET name=?,dept=?,gender=?,age=? WHERE reg=?;""",
        (nametxt.get(),depttxt.get(),a,w.get(),regtxt.get()))
```

Function defined to update existing values from the database

```
b2=Button(a,text='Update',command=update)
def delete():
    cur.execute("""DELETE FROM student WHERE reg=?;""",(regtxt.get(),))

b3=Button(a,text='Delete',command=delete)
```

Function defined to delete to the current value from the database

```
def select():
    cur.execute("""SELECT * FROM student WHERE reg=?;""",(regtxt.get(),))
    ans=cur.fetchall()
    for i in ans:
        print(ans)

b4=Button(a,text='Select',command=select)

b1.grid(column=0,row=5)
b2.grid(column=1,row=5)
b3.grid(column=0,row=6)
b4.grid(column=1,row=6)

con.commit()
#con.close()

a.mainloop()
```
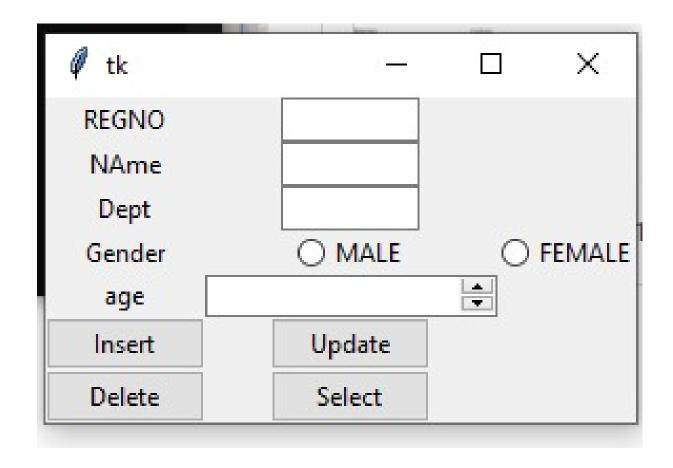
Function defined to select values from the table

Conclusion

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming.