

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in

Information Technology

By

NAME:ROHINI M
(Reg. No:38120067)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

JANUARY 2020



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -
600119 www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ROHINI M** (Reg.no.38120067) who carried out the project entitled “**CONTACT MANAGEMENT SYSTEM USING PYTHON**” under our supervision from Jan 2021 to April 2021.

Internal Guide

Name and Signature,

Dr.R.M.Gomathi,M.Tech.,Ph.D.

Head of the Department

Name and Signature

Dr.SUBHASHINI M.E., Ph.D. ,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **ROHINI M** hereby declare that the Project Report entitled “**CONTACT MANAGEMENT SYSTEM**”done by me under the guidance of **Dr.R.M.GOMATHI,M.Tech.,Ph.D.(Internal)** and **Dr.SUBHASHINI M.E., Ph.D. , (External)** at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in **INFORMATION TECHNOLOGY**

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, Dean, School of Computing **Dr.SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.R.M.GOMATHI,M.Tech.,Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

INTERNSHIP CERTIFICATION

**RETECH**
SOLUTIONS PVT. LTD.
An ISO 9001:2015 certified company

INT: RE20S036

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **ROHINI M** student of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, BTECH/IT** has undergone the Internship Training through online in our concern entitled **PYTHON** from **19th Aug 2020** to **21st Aug 2020** in relevant departments related to their academic studies.

During the above period, the performance was good & we wish great success in all your future endeavours.



Authorised Signature

Technology drives the world We are the people who are behind the technology.



No-31, 1st Floor, Alagesan Street, Tambaram West, Chennai-45.
Office : 044-42077204 | +91-8122266653 www.retechsolutions.in

ABSTRACT

Abstract of the project contact management system:

The purpose of contact management system is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. the required software and hardware are easily available and easy to work with.

Contact management system, as described above, can lead to error free, secure, reliable and fast management system.

It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organisation in better utilization of resources.

The organisation can maintain computerized records without redundant entries. That means that one need not to be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with ease accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

Objective of project on contact management system:

The main objective of the project on contact management system is to manage the details of Contact ,First Name, Last Name, Gender, Age, Contact Number ,Address.

The project is totally built at administrative end and thus only the administrator is guaranteed the access, The purpose of the project is to build an application program to reduce the manual work for managing the Contact, Credential, Mobile, Telephone. It tracks all the details about Telephone.

1.INTRODUCTION:

Contact Management System project is written in Python. The project file contains a python script (index.py). This is a simple GUI based project which is very easy to understand and use.

Talking about the system, it contains all the required functions which include adding, viewing, deleting and updating contact lists. While adding the contact of a person, he/she has to provide first name, last name, gender, address and contact details.

The user can also update the contact list if he/she wants to. For this, the user has to double-click on a record that he/she wishes to edit. The system shows the contact details in a list view. And also the user easily delete any contact details.

This GUI based Contact Management system provides the simplest management of contact details. In short, this projects mainly focus on CRUD. There's an external database connection file used in this mini project to save user's data permanently.

In order to run the project, you must have installed **Python**, on your PC. This is a simple GUI Based system, specially written for the beginners. Contact Management System in Python project with source code is free to download. Use for education purpose only!

2.AIM AND SCOPE

2.1 AIM:

On a daily basis, everyone depends on a lot of contacts which cannot be remembered so easily as it's tough to do so.

Hence dependability increases on phones. What will be the situation if our phone is lost and we do not easily remember our contacts and other important stuff? So I got this idea.

2.2 SCOPE:

- It may help collecting perfect management in details .In a very short time ,the collection will be obvious ,simple and sensible .It will help a person to know the management of past year perfectly and vividly.
- It also helps in current or works relative to Contact Management System .It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.
- Our project aim at business process automation ,we have tried to various processes of Contact Management System.

In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.

- In computer system ,it is not necessary to create the manifest .but we can directly print it ,which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation
- The system generates types of information that can be used for various purposes.
- It satisfy the user requirement.
- Be easy to understand by the user and operator.

- Be easy to operate
- Have a good user interface.
- Delivered on schedule within the budget.

3.SYSTEM ANALYSIS AND DESIGN

3.1 GENERAL

3.1.1 PYTHON

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

3.1.2 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

IDE stands for Integrated Development Environment. It's a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools,... And even though the IDE is a strictly defined concept, it's starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs. The IDE used in this project is Visual studio code.

3.1.3 USER INTERFACE

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website

4.Modules of Contact Management System:

- Contact Management Module: Used for managing the contact details .
- Mobile Module: Used for managing the detail of Mobile
- Address Module: Used for managing the detail of Address
- Gender Module: Used for managing the detail of Gender
- Age Module :Used for managing the detail of Age

5.Effective Methodology for Contact Management

The first and the most prominent step of establishing a contact management system is to engage directly with the customer who is seeking a solution to his/her problem through the product or service the company is providing. Contact management can be initiated by implementing one of the following ways

- You can directly engage with the customer by asking him about the products or services being promoted or advertised by the company.
- You can make cold calling, which includes calling up a particular person and asking him whether he or she might be interested in the new product or scheme inaugurated by the company.
- You can contact prospects by internal or external references, i.e., known acquaintances, relatives, or friends of a pre-existing customer.
- You can make prospects take online surveys and past track record of the web surfing of the customers.

- You can participate in online events and exhibition events where potential customers can give their details and express their perspectives about the product or service of a company.

When the lead has been created by acting upon the strategies mentioned above, it is then time to build on the lead.

It is strongly recommended to know as much as possible information about the preferences of a customer and try to acquire extensive knowledge about how much money potential customer is willing to spend. It will convert a customer into ever paying client.

Consistent engaging with the customers ensures the customer that an organization pays special heed on providing value-added incentives. Through direct interaction with the customer, company personnel can guide their respected customers towards the official website of a company, and utilize telephonic conversation to gain firsthand knowledge about the needs of the customers.

All the necessary information about a customer can be effectively stored in a way more robust way in a central database. Then another great way of knowing a personal preference of a customer is to keep the track record of IP addresses of the customers visiting the official website of a company regularly. Through this methodology, an organization can outreach more customers.

By acquiring a piece of more extensive knowledge about the needs of the customers, companies can emphasize significantly the sales and convince their customers to buy from them. It is often regarded as an excellent practice to make financial dealing in a face-to-face manner that leads to better interaction between the two parties.

Thereby your visitor has completely transformed into a customer. At this point, it is strongly recommended to register each customer with the company database.

The companies can manage to upsell their all products by accumulating customer feedback regularly. The organization can pay special heed to customer preference and psyche and make their product synchronize with customer needs.

6.Best Software for Contact Management

Automated software for contact management assists you in

keeping the track record of all the pre-existing customers. For the sake of the convenience of our readers, I'll be listing the 6 best software for contact management well-suited for small businesses.

1. Freshsales: This tool provides prospects' information who reach you via email, live chat, and web forms automatically. It provides your customer information in a simplified manner.
2. HubSpot CRM: This is a generalist tool for contact management includes the sales process and marketing. It alerts you by a notification when individual prospects visit your web site. It suits all types of businesses.
3. Pipedrive: This is a strongly automated, sales-oriented tool. This tool allows you to gather all sales data in one place and helps you to get all the information easily before a call or meeting.
4. Zoho CRM: This tool suits freelancers, small businesses, and startups. It provides a comprehensive view of communication through email, live chat, phone, etc.
5. Nimble: This tool helps you to have the best relationship with customers across all social media platforms. It automatically displays all the contact details and company details of customers including the details of communication you may have with them.
6. Metrillo:

Most of the contact management systems can be integrated with Enterprise resource planning systems of the organization.

7. Automated Process for the Contact Management

By automating the whole process of contact management, you'll be able to design a more robust contact management system that is more profitable, efficient, and cost-effective.

The automated system of contact management can define the interval at which you can call your potential customer and store all his personal information in a more retrievable manner.

After transforming the prospective visitor into a company customer, the system needs to perform the necessary changes to ensure the highest level of customer satisfaction.

. 8.Benefits of Contact Management

- orders to enthusiastic customers.

• Product and component based An automated system of contact management provides the following benefits to the customer:

- Addressing the customer concerns, and improve flaws in the product or service
- Companies can ensure the highest level of customer satisfaction by storing customer records in a retrievable manner with high efficiency and less overhead.
- Efficiently storing all the customer information from where it can be accessed by any personnel within the specific organization

Tracking the sales track record of a specific customer allows the company to promote interactive

- Creating & changing issues at ease
- Query issue list to any depth
- Reporting & charting in more comprehensive way
- User accounts to control the access and maintain security
- Simple status & resolutions
- Multi-level priorities & severities
- Targets & milestones for guiding the programmers
- Attachments & additional comments for more information
- Robust database back-end
- Various level of reports available with a lot of filter criteria's

- It contains better storage capacity
- Accuracy in work
- Easy & fast retrieval of information.
- Well designed reports.
- Decrease the load of the person involve in existing manual system.
- Access of any information individually
- Easy to update information

9.For a successful software project, the following steps can be followed:

- Select a project
 - o Identifying project's aims and objectives
 - o Understading requirements and specification
 - o Methods of analysis, design and implementation
 - o Testing techniques
 - o Documentation
- Project milestones and deliverables
- Budget allocation
 - o Exceeding limits within control
- Project estimates
 - o Cost
 - o Time
 - o Size of code
 - o Duration
- Resource allocation
 - o Hardware
 - o Software

- o Previous relevant project information
 - o Digital library
- Risk management
 - o Risk avoidance
 - o Risk detection

10.SUPPORT:

CONTACT MANAGEMENT:

800-755-7041 option 7 sub option 1

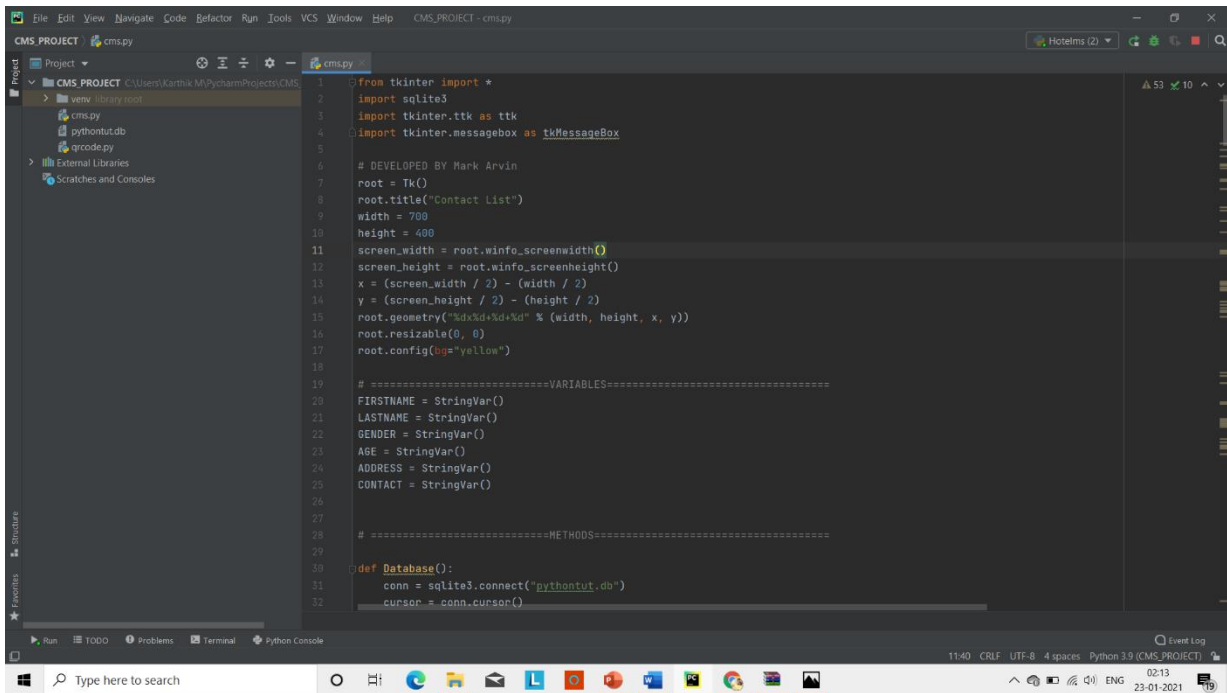
11.CONCLUSION:

So far, we've discussed all the critical aspects of the contact management automated system.

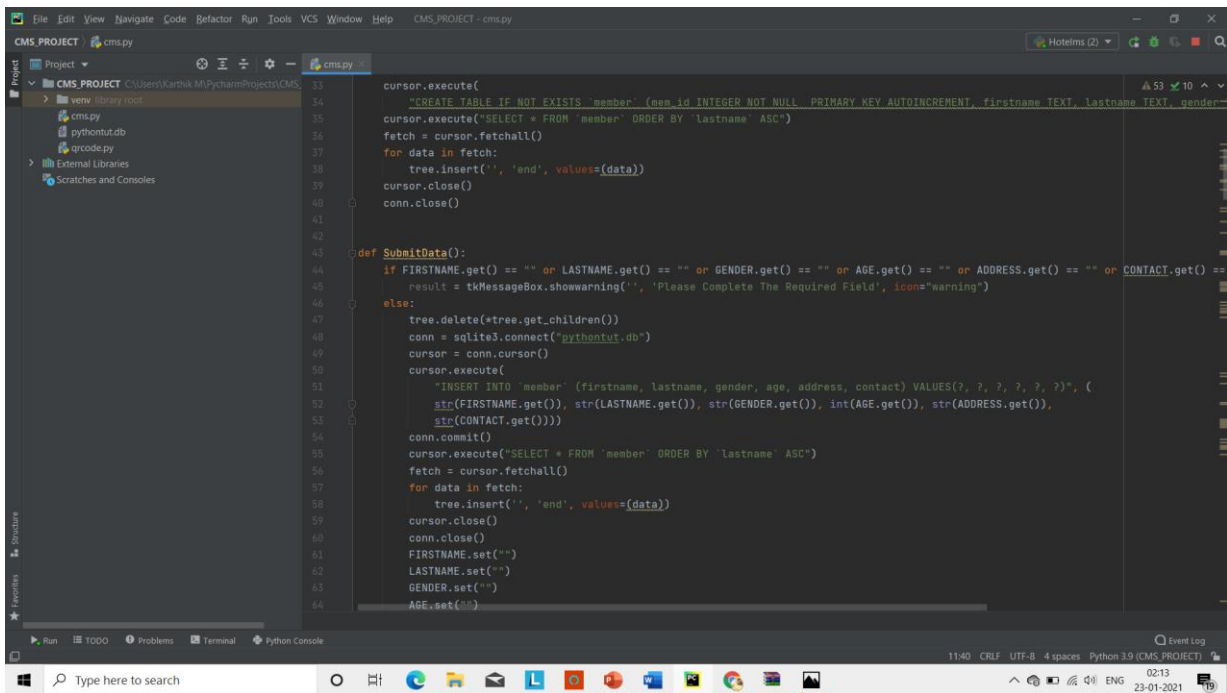
So far, whatever has been said, it should be taken into consideration that the success of any business, whether it is a large organization or a small startup solely relies on the level of satisfaction of the customer.

Just because of these reasons, large organizations emphasize significantly on the automated contact management system to reduce the company overhead and enhance their revenue as an outcome

CODING:



```
1 from tkinter import *
2 import sqlite3
3 import tkinter.ttk as ttk
4 import tkinter.messagebox as tkMessageBox
5
6 # DEVELOPED BY Mark Arvin
7 root = Tk()
8 root.title("Contact List")
9 width = 700
10 height = 400
11 screen_width = root.winfo_screenwidth()
12 screen_height = root.winfo_screenheight()
13 x = (screen_width / 2) - (width / 2)
14 y = (screen_height / 2) - (height / 2)
15 root.geometry("%dx%d+%d+%d" % (width, height, x, y))
16 root.resizable(0, 0)
17 root.config(bg="yellow")
18
19 # =====VARIABLES=====
20 FIRSTNAME = StringVar()
21 LASTNAME = StringVar()
22 GENDER = StringVar()
23 AGE = StringVar()
24 ADDRESS = StringVar()
25 CONTACT = StringVar()
26
27 # =====METHODS=====
28
29
30 def Database():
31     conn = sqlite3.connect("pythontut.db")
32     cursor = conn.cursor()
```



```
33 cursor.execute(
34     "CREATE TABLE IF NOT EXISTS 'member' (mem_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, firstname TEXT, lastname TEXT, gender"
35     cursor.execute("SELECT * FROM 'member' ORDER BY 'lastname' ASC")
36     fetch = cursor.fetchall()
37     for data in fetch:
38         tree.insert('', 'end', values=(data))
39     cursor.close()
40     conn.close()
41
42
43 def SubmitData():
44     if FIRSTNAME.get() == "" or LASTNAME.get() == "" or GENDER.get() == "" or AGE.get() == "" or ADDRESS.get() == "" or CONTACT.get() == "":
45         result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")
46     else:
47         tree.delete(*tree.get_children())
48         conn = sqlite3.connect("pythontut.db")
49         cursor = conn.cursor()
50         cursor.execute(
51             "INSERT INTO 'member' (firstname, lastname, gender, age, address, contact) VALUES(?, ?, ?, ?, ?, ?)", (
52                 str(FIRSTNAME.get()), str(LASTNAME.get()), str(GENDER.get()), int(AGE.get()), str(ADDRESS.get()),
53                 str(CONTACT.get()))
54         conn.commit()
55         cursor.execute("SELECT * FROM 'member' ORDER BY 'lastname' ASC")
56         fetch = cursor.fetchall()
57         for data in fetch:
58             tree.insert('', 'end', values=(data))
59         cursor.close()
60         conn.close()
61         FIRSTNAME.set("")
62         LASTNAME.set("")
63         GENDER.set("")
64         AGE.set("")
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help CMS_PROJECT - cms.py
CMS PROJECT cms.py
Project
  CMS PROJECT C:\Users\Kartik M\PycharmProjects\CMS
  > venv library root
  cms.py
  pythontut.db
  qrcode.py
  External Libraries
  Scratches and Consoles
  Run
  TODO
  Problems
  Terminal
  Python Console
  11:40 CRLF UTF-8 4 spaces Python 3.9 (CMS_PROJECT)
  02:14 23-01-2021

65 ADDRESS.set("")
66 CONTACT.set("")
67
68
69 def UpdateData():
70     if GENDER.get() == "":
71         result = tkMessageBox.showwarning('Please Complete The Required Field', icon="warning")
72     else:
73         tree.delete(*tree.get_children())
74         conn = sqlite3.connect("pythontut.db")
75         cursor = conn.cursor()
76         cursor.execute(
77             "UPDATE 'member' SET 'firstname' = ?, 'lastname' = ?, 'gender' = ?, 'age' = ?, 'address' = ?, 'contact' = ? WHERE 'mem_id' = ?"
78             (str(FIRSTNAME.get()), str(LASTNAME.get()), str(GENDER.get()), str(AGE.get()), str(ADDRESS.get()),
79              str(CONTACT.get()), int(mem_id)))
80         conn.commit()
81         cursor.execute("SELECT * FROM 'member' ORDER BY 'lastname' ASC")
82         fetch = cursor.fetchall()
83         for data in fetch:
84             tree.insert('', 'end', values=(data))
85         cursor.close()
86         conn.close()
87         FIRSTNAME.set("")
88         LASTNAME.set("")
89         GENDER.set("")
90         AGE.set("")
91         ADDRESS.set("")
92         CONTACT.set("")
93
94
95 def OnSelected(event):
96     global mem_id, UpdateWindow
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help CMS_PROJECT - cms.py
CMS PROJECT cms.py
Project
  CMS PROJECT C:\Users\Kartik M\PycharmProjects\CMS
  > venv library root
  cms.py
  pythontut.db
  qrcode.py
  External Libraries
  Scratches and Consoles
  Run
  TODO
  Problems
  Terminal
  Python Console
  11:40 CRLF UTF-8 4 spaces Python 3.9 (CMS_PROJECT)
  02:14 23-01-2021

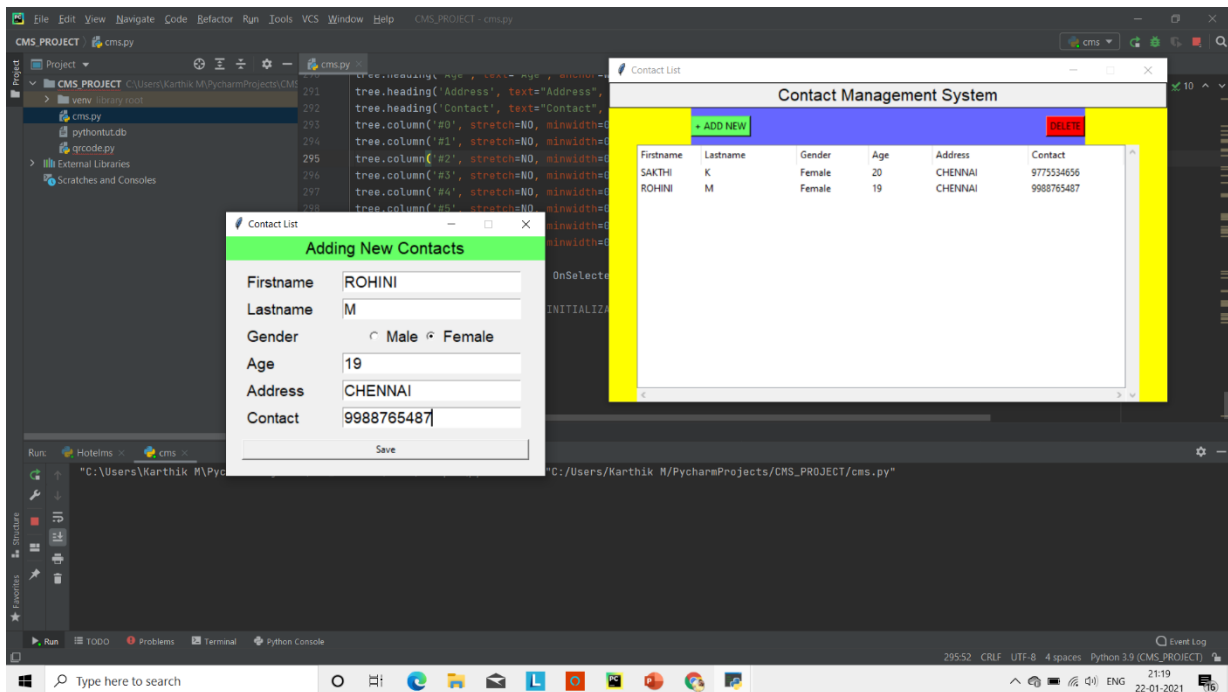
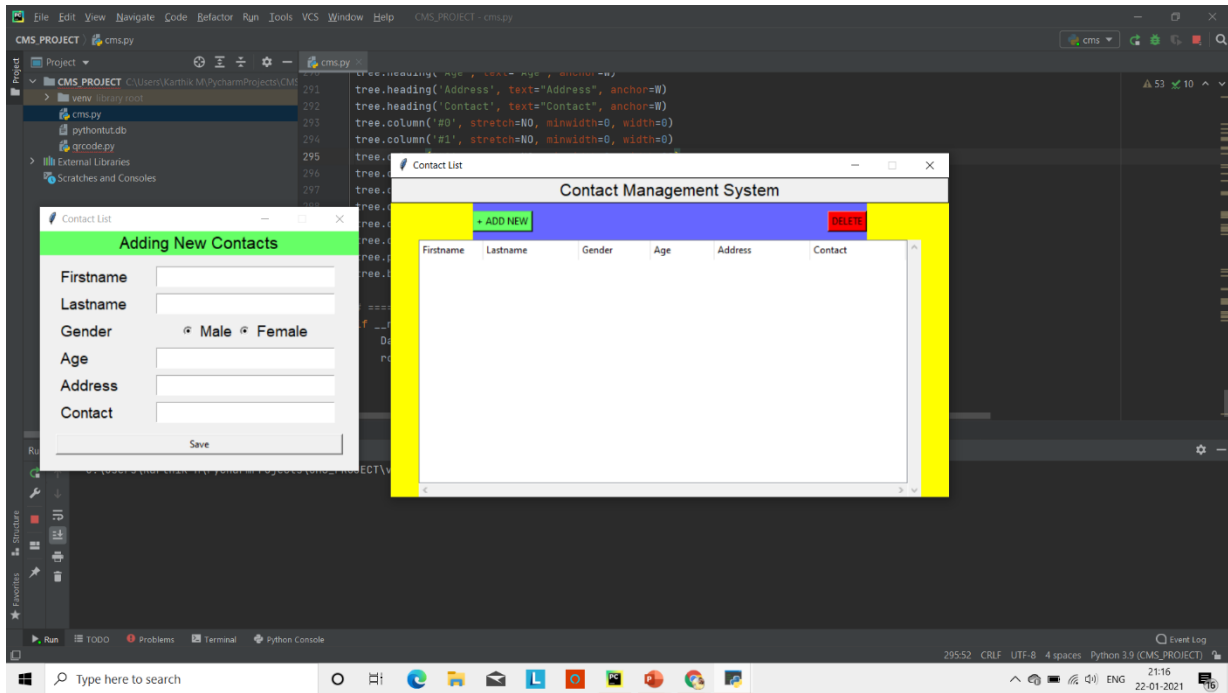
96 global mem_id, UpdateWindow
97 curitem = tree.focus()
98 contents = (tree.item(curitem))
99 selecteditem = contents['values']
100 mem_id = selecteditem[0]
101 FIRSTNAME.set("")
102 LASTNAME.set("")
103 GENDER.set("")
104 AGE.set("")
105 ADDRESS.set("")
106 CONTACT.set("")
107 FIRSTNAME.set(selecteditem[1])
108 LASTNAME.set(selecteditem[2])
109 AGE.set(selecteditem[3])
110 ADDRESS.set(selecteditem[4])
111 CONTACT.set(selecteditem[5])
112 UpdateWindow = Toplevel()
113 UpdateWindow.title("Contact List")
114 width = 400
115 height = 300
116 screen_width = root.winfo_screenwidth()
117 screen_height = root.winfo_screenheight()
118 x = ((screen_width / 2) + 450) - (width / 2)
119 y = ((screen_height / 2) + 20) - (height / 2)
120 UpdateWindow.resizable(0, 0)
121 UpdateWindow.geometry("%dx%d+%d+%d" % (width, height, x, y))
122 if 'NewWindow' in globals():
123     NewWindow.destroy()
124
125 # =====FRAMES=====
126 FormTitle = Frame(UpdateWindow)
127 FormTitle.pack(side=TOP)
```

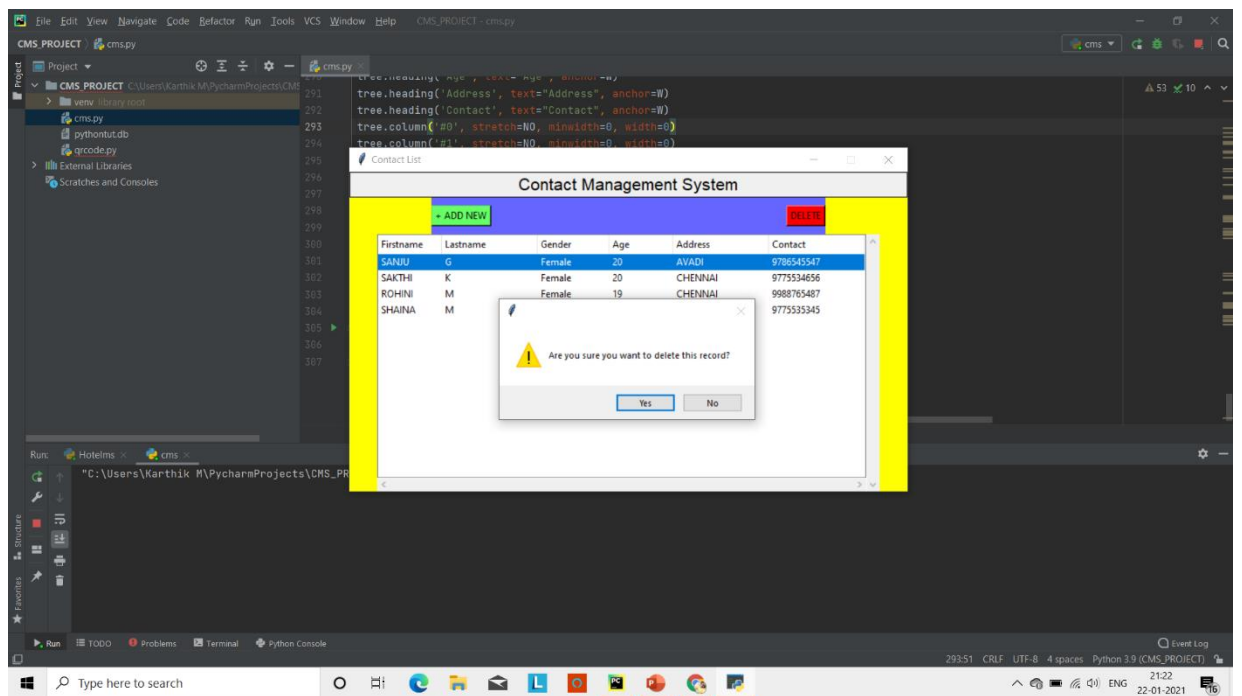
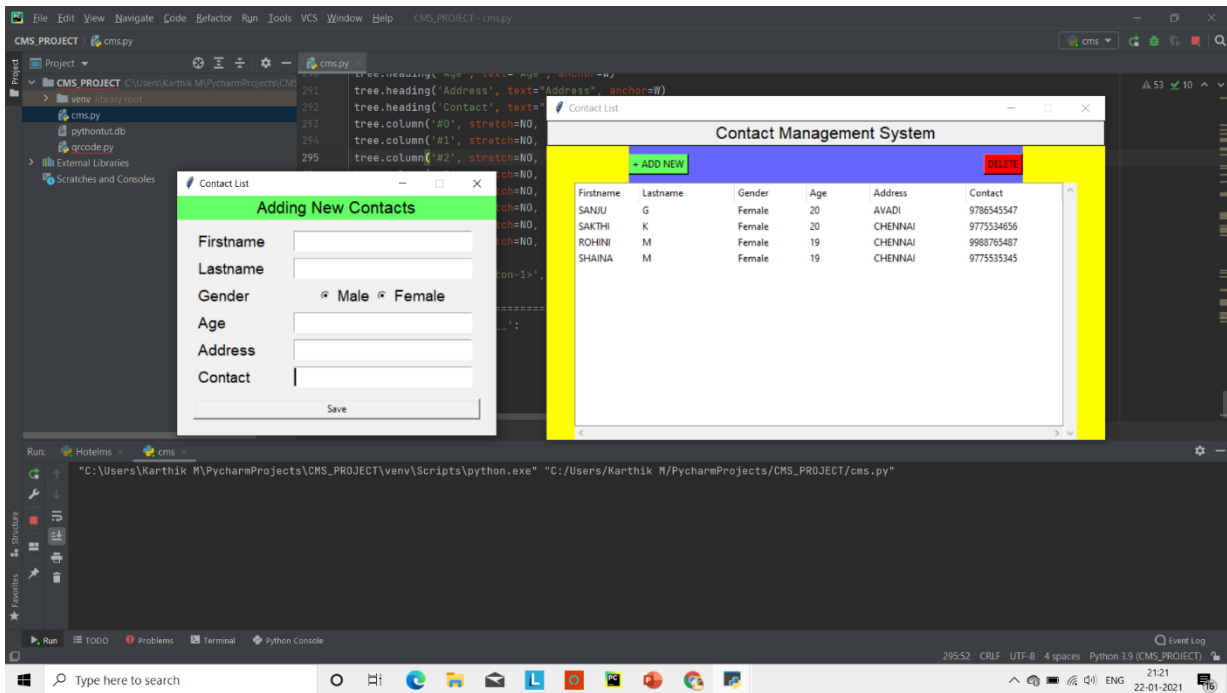


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help CMS_PROJECT - cms.py
CMS PROJECT C:\Users\Kartik M\PycharmProjects\CMS
Project
  CMS PROJECT
  cms.py
  pythonut.db
  qrcode.py
  External Libraries
  Scratches and Consoles
  cms.py
191 GENDER.set("")
192 AGE.set("")
193 ADDRESS.set("")
194 CONTACT.set("")
195 NewWindow = Toplevel()
196 NewWindow.title("Contact List")
197 width = 400
198 height = 300
199 screen_width = root.winfo_screenwidth()
200 screen_height = root.winfo_screenheight()
201 x = ((screen_width / 2) - 455) - (width / 2)
202 y = ((screen_height / 2) + 20) - (height / 2)
203 NewWindow.resizable(0, 0)
204 NewWindow.geometry("%dx%d+%d+%d" % (width, height, x, y))
205 if 'UpdateWindow' in globals():
206     UpdateWindow.destroy()
207
208 # =====FRAMES=====
209 FormTitle = Frame(NewWindow)
210 FormTitle.pack(side=TOP)
211 ContactForm = Frame(NewWindow)
212 ContactForm.pack(side=TOP, pady=10)
213 RadioGroup = Frame(ContactForm)
214 Male = Radiobutton(RadioGroup, text="Male", variable=GENDER, value="Male", font=('arial', 14)).pack(side=LEFT)
215 Female = Radiobutton(RadioGroup, text="Female", variable=GENDER, value="Female", font=('arial', 14)).pack(side=LEFT)
216
217 # =====LABELS=====
218 lbl_title = Label(FormTitle, text="Adding New Contacts", font=('arial', 16), bg="#6666ff", width=300)
219 lbl_title.pack(fill=X)
220 lbl_firstname = Label(ContactForm, text="Firstname", font=('arial', 14), bd=5)
221 lbl_firstname.grid(row=0, sticky=W)
222 lbl_lastname = Label(ContactForm, text="Lastname", font=('arial', 14), bd=5)
223
224 AddNewWindow()
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help CMS_PROJECT - cms.py
CMS PROJECT C:\Users\Kartik M\PycharmProjects\CMS
Project
  CMS PROJECT
  cms.py
  pythonut.db
  qrcode.py
  External Libraries
  Scratches and Consoles
  cms.py
254 Mid = Frame(root, width=500, bg="#6666ff")
255 Mid.pack(side=TOP)
256 MidLeft = Frame(Mid, width=100)
257 MidLeft.pack(side=LEFT, pady=10)
258 MidLeftPadding = Frame(Mid, width=370, bg="#6666ff")
259 MidLeftPadding.pack(side=LEFT)
260 MidRight = Frame(Mid, width=100)
261 MidRight.pack(side=RIGHT, pady=10)
262 TableMargin = Frame(root, width=500)
263 TableMargin.pack(side=TOP)
264 # =====LABELS=====
265 lbl_title = Label(Top, text="Contact Management System", font=('arial', 16), width=500)
266 lbl_title.pack(fill=X)
267
268 # =====ENTRY=====
269
270 # =====BUTTONS=====
271 btn_add = Button(MidLeft, text="ADD NEW", bg="#6666ff", command=AddNewWindow)
272 btn_add.pack()
273 btn_delete = Button(MidRight, text="DELETE", bg="red", command>DeleteData)
274 btn_delete.pack(side=RIGHT)
275
276 # =====TABLES=====
277 scrollbar = Scrollbar(TableMargin, orient=HORIZONTAL)
278 scrollbar.pack(side=RIGHT, fill=Y)
279 tree = ttk.Treeview(TableMargin, columns=("MemberID", "Firstname", "Lastname", "Gender", "Age", "Address", "Contact"),
280 height=400, selectmode="extended", yscrollcommand=scrollbar.set, xscrollcommand=scrollbarx.set)
281 scrollbar.config(command=tree.yview)
282 scrollbar.pack(side=RIGHT, fill=Y)
283 scrollbarx.config(command=tree.xview)
284 scrollbarx.pack(side=BOTTOM, fill=X)
285 # =====HEADINGS=====
286 AddNewWindow()
```

OUTPUT:





APENDICES:

```
from tkinter import *
import sqlite3
import tkinter.ttk as ttk
import tkinter.messagebox as tkMessageBox
```

```
#DEVELOPED BY Mark Arvin
```

```
root = Tk()
root.title("Contact List")
width = 700
height = 400
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
x = (screen_width/2) - (width/2)
y = (screen_height/2) - (height/2)
root.geometry("%dx%d+%d+%d" % (width, height, x, y))
root.resizable(0, 0)
root.config(bg="#6666ff")
```

```
#=====VARIABLES=====
=====
```

```
FIRSTNAME = StringVar()
LASTNAME = StringVar()
GENDER = StringVar()
AGE = StringVar()
ADDRESS = StringVar()
CONTACT = StringVar()
```

```
#=====METHODS=====
=====
```

```
def Database():
    conn = sqlite3.connect("pythontut.db")
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS
`member` (mem_id INTEGER NOT NULL PRIMARY KEY
AUTOINCREMENT, firstname TEXT, lastname TEXT, gender
```

```

TEXT, age TEXT, address TEXT, contact TEXT)")
    cursor.execute("SELECT * FROM `member` ORDER BY
`lastname` ASC")
    fetch = cursor.fetchall()
    for data in fetch:
        tree.insert("", 'end', values=(data))
    cursor.close()
    conn.close()

```

```

def SubmitData():
    if FIRSTNAME.get() == "" or LASTNAME.get() == "" or
    GENDER.get() == "" or AGE.get() == "" or ADDRESS.get() ==
    "" or CONTACT.get() == "":
        result = tkMessageBox.showwarning("", 'Please
Complete The Required Field', icon="warning")
    else:
        tree.delete(*tree.get_children())
        conn = sqlite3.connect("pythontut.db")
        cursor = conn.cursor()
        cursor.execute("INSERT INTO `member` (firstname,
lastname, gender, age, address, contact)
VALUES(?, ?, ?, ?, ?, ?)", (str(FIRSTNAME.get()),
str(LASTNAME.get()), str(GENDER.get()), int(AGE.get()),
str(ADDRESS.get()), str(CONTACT.get()))
        conn.commit()
        cursor.execute("SELECT * FROM `member` ORDER BY
`lastname` ASC")
        fetch = cursor.fetchall()
        for data in fetch:
            tree.insert("", 'end', values=(data))
        cursor.close()
        conn.close()
        FIRSTNAME.set("")
        LASTNAME.set("")
        GENDER.set("")
        AGE.set("")
        ADDRESS.set("")
        CONTACT.set("")

```

```

def UpdateData():

```

```

if GENDER.get() == "":
    result = tkMessageBox.showwarning("", 'Please
Complete The Required Field', icon="warning")
else:
    tree.delete(*tree.get_children())
    conn = sqlite3.connect("pythontut.db")
    cursor = conn.cursor()
    cursor.execute("UPDATE `member` SET `firstname`
= ?, `lastname` = ?, `gender` = ?, `age` = ?, `address` = ?,
`contact` = ? WHERE `mem_id` = ?", (str(FIRSTNAME.get()),
str(LASTNAME.get()), str(GENDER.get()), str(AGE.get()),
str(ADDRESS.get()), str(CONTACT.get()), int(mem_id)))
    conn.commit()
    cursor.execute("SELECT * FROM `member` ORDER BY
`lastname` ASC")
    fetch = cursor.fetchall()
    for data in fetch:
        tree.insert("", 'end', values=(data))
    cursor.close()
    conn.close()
    FIRSTNAME.set("")
    LASTNAME.set("")
    GENDER.set("")
    AGE.set("")
    ADDRESS.set("")
    CONTACT.set("")

```

```

def OnSelected(event):
    global mem_id, UpdateWindow
    curItem = tree.focus()
    contents =(tree.item(curItem))
    selecteditem = contents['values']
    mem_id = selecteditem[0]
    FIRSTNAME.set("")
    LASTNAME.set("")
    GENDER.set("")
    AGE.set("")
    ADDRESS.set("")
    CONTACT.set("")

```

```

FIRSTNAME.set(selecteditem[1])
LASTNAME.set(selecteditem[2])
AGE.set(selecteditem[4])
ADDRESS.set(selecteditem[5])
CONTACT.set(selecteditem[6])
UpdateWindow = Toplevel()
UpdateWindow.title("Contact List")
width = 400
height = 300
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
x = ((screen_width/2) + 450) - (width/2)
y = ((screen_height/2) + 20) - (height/2)
UpdateWindow.resizable(0, 0)
UpdateWindow.geometry("%dx%d+%d+%d" % (width,
height, x, y))
if 'NewWindow' in globals():
    NewWindow.destroy()

#=====FRAMES=====
=====
FormTitle = Frame(UpdateWindow)
FormTitle.pack(side=TOP)
ContactForm = Frame(UpdateWindow)
ContactForm.pack(side=TOP, pady=10)
RadioGroup = Frame(ContactForm)
Male = Radiobutton(RadioGroup, text="Male",
variable=GENDER, value="Male", font=('arial',
14)).pack(side=LEFT)
Female = Radiobutton(RadioGroup, text="Female",
variable=GENDER, value="Female", font=('arial',
14)).pack(side=LEFT)

#=====LABELS=====
=====
lbl_title = Label(FormTitle, text="Updating Contacts",
font=('arial', 16), bg="orange", width = 300)
lbl_title.pack(fill=X)
lbl_firstname = Label(ContactForm, text="Firstname",
font=('arial', 14), bd=5)

```

```

    lbl_firstname.grid(row=0, sticky=W)
    lbl_lastname = Label(ContactForm, text="Lastname",
font=('arial', 14), bd=5)
    lbl_lastname.grid(row=1, sticky=W)
    lbl_gender = Label(ContactForm, text="Gender",
font=('arial', 14), bd=5)
    lbl_gender.grid(row=2, sticky=W)
    lbl_age = Label(ContactForm, text="Age", font=('arial',
14), bd=5)
    lbl_age.grid(row=3, sticky=W)
    lbl_address = Label(ContactForm, text="Address",
font=('arial', 14), bd=5)
    lbl_address.grid(row=4, sticky=W)
    lbl_contact = Label(ContactForm, text="Contact",
font=('arial', 14), bd=5)
    lbl_contact.grid(row=5, sticky=W)

#=====ENTRY=====
=====
    firstname = Entry(ContactForm, textvariable=FIRSTNAME,
font=('arial', 14))
    firstname.grid(row=0, column=1)
    lastname = Entry(ContactForm, textvariable=LASTNAME,
font=('arial', 14))
    lastname.grid(row=1, column=1)
    RadioGroup.grid(row=2, column=1)
    age = Entry(ContactForm, textvariable=AGE, font=('arial',
14))
    age.grid(row=3, column=1)
    address = Entry(ContactForm,
textvariable=ADDRESS, font=('arial', 14))
    address.grid(row=4, column=1)
    contact = Entry(ContactForm,
textvariable=CONTACT, font=('arial', 14))
    contact.grid(row=5, column=1)

#=====BUTTONS=====
=====
    btn_updatecon = Button(ContactForm, text="Update",

```

```
width=50, command=UpdateData)
    btn_updatecon.grid(row=6, columnspan=2, pady=10)
```

```
#fn1353p
```

```
def DeleteData():
```

```
    if not tree.selection():
```

```
        result = tkMessageBox.showwarning("", 'Please Select  
Something First!', icon="warning")
```

```
    else:
```

```
        result = tkMessageBox.askquestion("", 'Are you sure  
you want to delete this record?', icon="warning")
```

```
        if result == 'yes':
```

```
            curltem = tree.focus()
```

```
            contents =(tree.item(curltem))
```

```
            selecteditem = contents['values']
```

```
            tree.delete(curltem)
```

```
            conn = sqlite3.connect("pythontut.db")
```

```
            cursor = conn.cursor()
```

```
            cursor.execute("DELETE FROM `member` WHERE  
`mem_id` = %d" % selecteditem[0])
```

```
            conn.commit()
```

```
            cursor.close()
```

```
            conn.close()
```

```
def AddNewWindow():
```

```
    global NewWindow
```

```
    FIRSTNAME.set("")
```

```
    LASTNAME.set("")
```

```
    GENDER.set("")
```

```
    AGE.set("")
```

```
    ADDRESS.set("")
```

```
    CONTACT.set("")
```

```
    NewWindow = Toplevel()
```

```
    NewWindow.title("Contact List")
```

```
    width = 400
```

```
    height = 300
```

```
    screen_width = root.winfo_screenwidth()
```

```
    screen_height = root.winfo_screenheight()
```

```
    x = ((screen_width/2) - 455) - (width/2)
```

```

y = ((screen_height/2) + 20) - (height/2)
NewWindow.resizable(0, 0)
NewWindow.geometry("%dx%d+%d+%d" % (width,
height, x, y))
if 'UpdateWindow' in globals():
    UpdateWindow.destroy()

#=====FRAMES=====
=====
FormTitle = Frame(NewWindow)
FormTitle.pack(side=TOP)
ContactForm = Frame(NewWindow)
ContactForm.pack(side=TOP, pady=10)
RadioGroup = Frame(ContactForm)
Male = Radiobutton(RadioGroup, text="Male",
variable=GENDER, value="Male", font=('arial',
14)).pack(side=LEFT)
Female = Radiobutton(RadioGroup, text="Female",
variable=GENDER, value="Female", font=('arial',
14)).pack(side=LEFT)

#=====LABELS=====
=====
lbl_title = Label(FormTitle, text="Adding New Contacts",
font=('arial', 16), bg="#66ff66", width = 300)
lbl_title.pack(fill=X)
lbl_firstname = Label(ContactForm, text="Firstname",
font=('arial', 14), bd=5)
lbl_firstname.grid(row=0, sticky=W)
lbl_lastname = Label(ContactForm, text="Lastname",
font=('arial', 14), bd=5)
lbl_lastname.grid(row=1, sticky=W)
lbl_gender = Label(ContactForm, text="Gender",
font=('arial', 14), bd=5)
lbl_gender.grid(row=2, sticky=W)
lbl_age = Label(ContactForm, text="Age", font=('arial',
14), bd=5)
lbl_age.grid(row=3, sticky=W)
lbl_address = Label(ContactForm, text="Address",
font=('arial', 14), bd=5)

```

```

    lbl_address.grid(row=4, sticky=W)
    lbl_contact = Label(ContactForm, text="Contact",
font=('arial', 14), bd=5)
    lbl_contact.grid(row=5, sticky=W)

#=====ENTRY=====
=====
    firstname = Entry(ContactForm, textvariable=FIRSTNAME,
font=('arial', 14))
    firstname.grid(row=0, column=1)
    lastname = Entry(ContactForm, textvariable=LASTNAME,
font=('arial', 14))
    lastname.grid(row=1, column=1)
    RadioGroup.grid(row=2, column=1)
    age = Entry(ContactForm, textvariable=AGE, font=('arial',
14))
    age.grid(row=3, column=1)
    address = Entry(ContactForm,
textvariable=ADDRESS, font=('arial', 14))
    address.grid(row=4, column=1)
    contact = Entry(ContactForm,
textvariable=CONTACT, font=('arial', 14))
    contact.grid(row=5, column=1)

#=====BUTTONS=====
=====
    btn_addcon = Button(ContactForm, text="Save",
width=50, command=SubmitData)
    btn_addcon.grid(row=6, columnspan=2, pady=10)

#=====FRAMES=====
=====
Top = Frame(root, width=500, bd=1, relief=SOLID)
Top.pack(side=TOP)
Mid = Frame(root, width=500, bg="#6666ff")

```

```

Mid.pack(side=TOP)
MidLeft = Frame(Mid, width=100)
MidLeft.pack(side=LEFT, pady=10)
MidLeftPadding = Frame(Mid, width=370, bg="#6666ff")
MidLeftPadding.pack(side=LEFT)
MidRight = Frame(Mid, width=100)
MidRight.pack(side=RIGHT, pady=10)
TableMargin = Frame(root, width=500)
TableMargin.pack(side=TOP)
#=====LABELS=====
=====
lbl_title = Label(Top, text="Contact Management System",
font=('arial', 16), width=500)
lbl_title.pack(fill=X)

#=====ENTRY=====
=====

#=====BUTTONS=====
=====
btn_add = Button(MidLeft, text="+ ADD NEW", bg="#66ff66",
command=AddNewWindow)
btn_add.pack()
btn_delete = Button(MidRight, text="DELETE", bg="red",
command>DeleteData)
btn_delete.pack(side=RIGHT)

#=====TABLES=====
=====
scrollbarx = Scrollbar(TableMargin, orient=HORIZONTAL)
scrollbary = Scrollbar(TableMargin, orient=VERTICAL)
tree = ttk.Treeview(TableMargin, columns=("MemberID",
"Firstname", "Lastname", "Gender", "Age", "Address",
"Contact"), height=400, selectmode="extended",
yscrollcommand=scrollbary.set,
xscrollcommand=scrollbarx.set)
scrollbary.config(command=tree.yview)
scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)

```

```

tree.heading('MemberID', text="MemberID", anchor=W)
tree.heading('Firstname', text="Firstname", anchor=W)
tree.heading('Lastname', text="Lastname", anchor=W)
tree.heading('Gender', text="Gender", anchor=W)
tree.heading('Age', text="Age", anchor=W)
tree.heading('Address', text="Address", anchor=W)
tree.heading('Contact', text="Contact", anchor=W)
tree.column('#0', stretch=NO, minwidth=0, width=0)
tree.column('#1', stretch=NO, minwidth=0, width=0)
tree.column('#2', stretch=NO, minwidth=0, width=80)
tree.column('#3', stretch=NO, minwidth=0, width=120)
tree.column('#4', stretch=NO, minwidth=0, width=90)
tree.column('#5', stretch=NO, minwidth=0, width=80)
tree.column('#6', stretch=NO, minwidth=0, width=120)
tree.column('#7', stretch=NO, minwidth=0, width=120)
tree.pack()
tree.bind('<Double-Button-1>', OnSelected)

```

```

#=====INITIALIZATION=====
=====
if __name__ == '__main__':
    Database()
    root.mainloop()

```