**BIGLEARN TRINING INSTITUTE**

**TRICHY**



**STUDENT MANAGEMENT SYSTEM**

**GUIDED BY: BY**

**V.AUGNES MARRY PRADEEP.E.G.**

**INTRODUCTION:**

Creating an address book in Python using Tkinter is an excellent project to learn GUI development while managing and organizing contact information efficiently. Tkinter, Python's standard GUI library, simplifies the process of designing interactive and user-friendly applications. The address book application allows users to store, view, edit, and delete contact details, such as names, phone numbers, and addresses. By combining Tkinter for the interface and Python's robust data handling capabilities, this project demonstrates essential programming concepts like event handling, database management, and user input validation. It's an ideal beginner-friendly project for enhancing programming and GUI development skills.

**FEATURES :**

Creating an address book in Python using Tkinter offers numerous features to streamline contact management. Users can add, view, edit, and delete contact details such as names, phone numbers, and addresses through a user-friendly graphical interface. The application supports searching for specific contacts quickly, ensuring efficient access to information. Data persistence can be achieved using a file or database to store contact details securely. The interface is designed for simplicity and responsiveness, catering to both beginners and experienced users. Additional features like sorting contacts alphabetically, exporting data, or adding profile pictures can enhance functionality, making it versatile and practical.

**SYSTEM CONFIGURATION**

PROCESSOR 11TH GEN Intel Core(R) Core(TM)

Installed RAM 8.0GB

System type 64-bit os , x64-based processor

SYSTEM REQUIREMENTS:

1.PYTHON(IDLE)

**PYTHON PACKAGES USED :**

TKINTER

**PROGRAM:**

# Import Module

from tkinter import \*

# Create Object

root = Tk()

# Set geometry

root.geometry('400x500')

# Information List

datas = []

# Add Information

def add():

global datas

datas.append([Name.get(),Number.get(),address.get(1.0, "end-1c")])

update\_book()

# View Information

def view():

Name.set(datas[int(select.curselection()[0])][0])

Number.set(datas[int(select.curselection()[0])][1])

address.delete(1.0,"end")

address.insert(1.0, datas[int(select.curselection()[0])][2])

# Delete Information

def delete():

del datas[int(select.curselection()[0])]

update\_book()

def reset():

Name.set('')

Number.set('')

address.delete(1.0,"end")

# Update Information

def update\_book():

select.delete(0,END)

for n,p,a in datas:

select.insert(END, n)

# Add Buttons, Label, ListBox

Name = StringVar()

Number = StringVar()

frame = Frame()

frame.pack(pady=10)

frame1 = Frame()

frame1.pack()

frame2 = Frame()

frame2.pack(pady=10)

Label(frame, text = 'Name', font='arial 12 bold').pack(side=LEFT)

Entry(frame, textvariable = Name,width=50).pack()

Label(frame1, text = 'Phone No.', font='arial 12 bold').pack(side=LEFT)

Entry(frame1, textvariable = Number,width=50).pack()

Label(frame2, text = 'Address', font='arial 12 bold').pack(side=LEFT)

address = Text(frame2,width=37,height=10)

address.pack()

Button(root,text="Add",font="arial 12 bold",command=add).place(x= 100, y=270)

Button(root,text="View",font="arial 12 bold",command=view).place(x= 100, y=310)

Button(root,text="Delete",font="arial 12 bold",command=delete).place(x= 100, y=350)

Button(root,text="Reset",font="arial 12 bold",command=reset).place(x= 100, y=390)

scroll\_bar = Scrollbar(root, orient=VERTICAL)

select = Listbox(root, yscrollcommand=scroll\_bar.set, height=12)

scroll\_bar.config (command=select.yview)

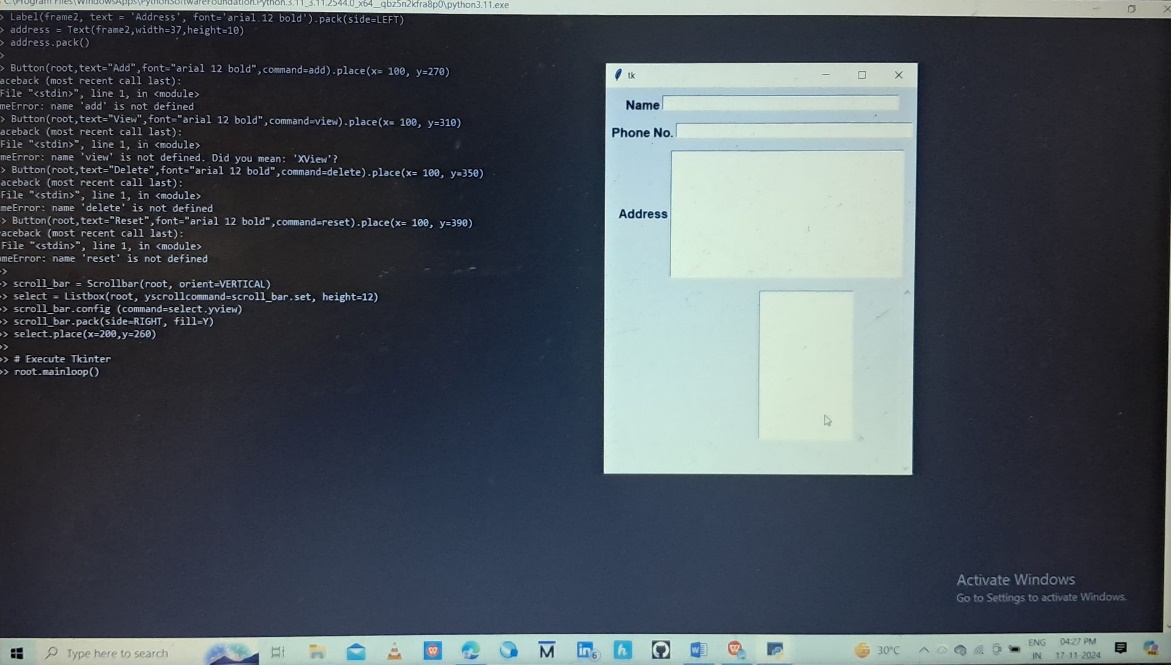
scroll\_bar.pack(side=RIGHT, fill=Y)

select.place(x=200,y=260)

# Execute Tkinter

root.mainloop()

**OUTPUT:**



**CONCLUSION:**

In conclusion, creating an address book in Python using Tkinter combines functionality with simplicity, making it a valuable project for learning GUI development. It provides essential features for managing contacts effectively while showcasing Python’s versatility. This project is an excellent stepping stone for developing more advanced and user-friendly applications in the future.