CHAPTER 1

INTRODUCTION

Charity is an act of kindness where a person who has more than enough of what he or she needs contributes a part of his or her surplus income for the fulfilment of the needs of those who are less capable. The mini project is basically providing the user to take a step forward and voluntarily help the people who are in need. Helping the needy is always a good deed. The main objective of the project is to allow the user to donate money, take the money from the user, and successfully send it to the organization or to the families to whom the user has donated the amount. It will help the people to raise funds of families of the martyred soldiers or help the flood-affected victims who have lost their homes, families etc. It can also help the government so that they can take a step and, with the money donated by us to them, help those who have lost their homes and are waiting for some help to arrive. It can be a flexible and easy way for a person to donate some amount for those who are in need.

1.1 COURSE OBJECTIVES

From this course, we will have a short research on the library provided by Python for creation of a Graphical User Interface (GUI) i.e., Tkinter. We also can learn about the various widgets which are provided by Tkinter for the creation of the GUI. Tkinter is a very easy ad user-friendly way of creating a GUI. We can also have a glance about MySQL, which is the database that is being used in this project. MySQL plays a very important role in storage management in this project. We also can learn how to connect MySQL to Tkinter and how to store the data entered in the GUI created to the MySQL database.

1.2 PROBLEM DEFINITION

The problem statement is basically helping the transfer of money take place from a donor to the soldier or the government body he/she wants to donate to. Through this GUI, we are helping the users to take an iniative to help those who are in need of money i.e., the families whose soldiers have been martyred and families who have been affected brutally by the floods. The GUI created in this project provides two options to the user whether he/she wants to donate to the soldiers who have sacrificed their life while protecting our beloved nation or to donate money to the flood-affected victims.

They have been given the facility to enter their bank details and pay their money via Internet Banking to directly send the money either into the soldier's account or into the account of the flood-affected victims.

1.3 PROJECT OUTCOMES

The project provides options to the user as to whom they want to donate their money to. Once they select the option, the GUI for the particular option opens and if the user selects option 1 i.e., if he wants to donate money to the families of martyred soldiers, the GUI will contain a list of names of the martyred soldiers. When option 1 is clicked, a drop-down menu appears which contains a list of all the soldiers who have sacrificed their life for this nation. Once a soldier is selected, his bank details will appear ion the screen and the user will have to enter his/her bank details and the amount he/she wants to donate to that soldier. After filling the details and clicking Procced, the money will be transferred to the soldier's account. If Option 2 is selected, a drop-down menu which contains a list of the Chief Minister's Relief Fund of various states. Once an option is selected, the bank details of that relief fund is displayed and after entering the bank details and clicking on proceed, the money will be transferred from donor's account to the Relief Fund of that particular state.

CHAPTER 2

REQUIREMENTS AND DESIGN

The requirements of the project are as follows:

2.1 Hardware Requirements:

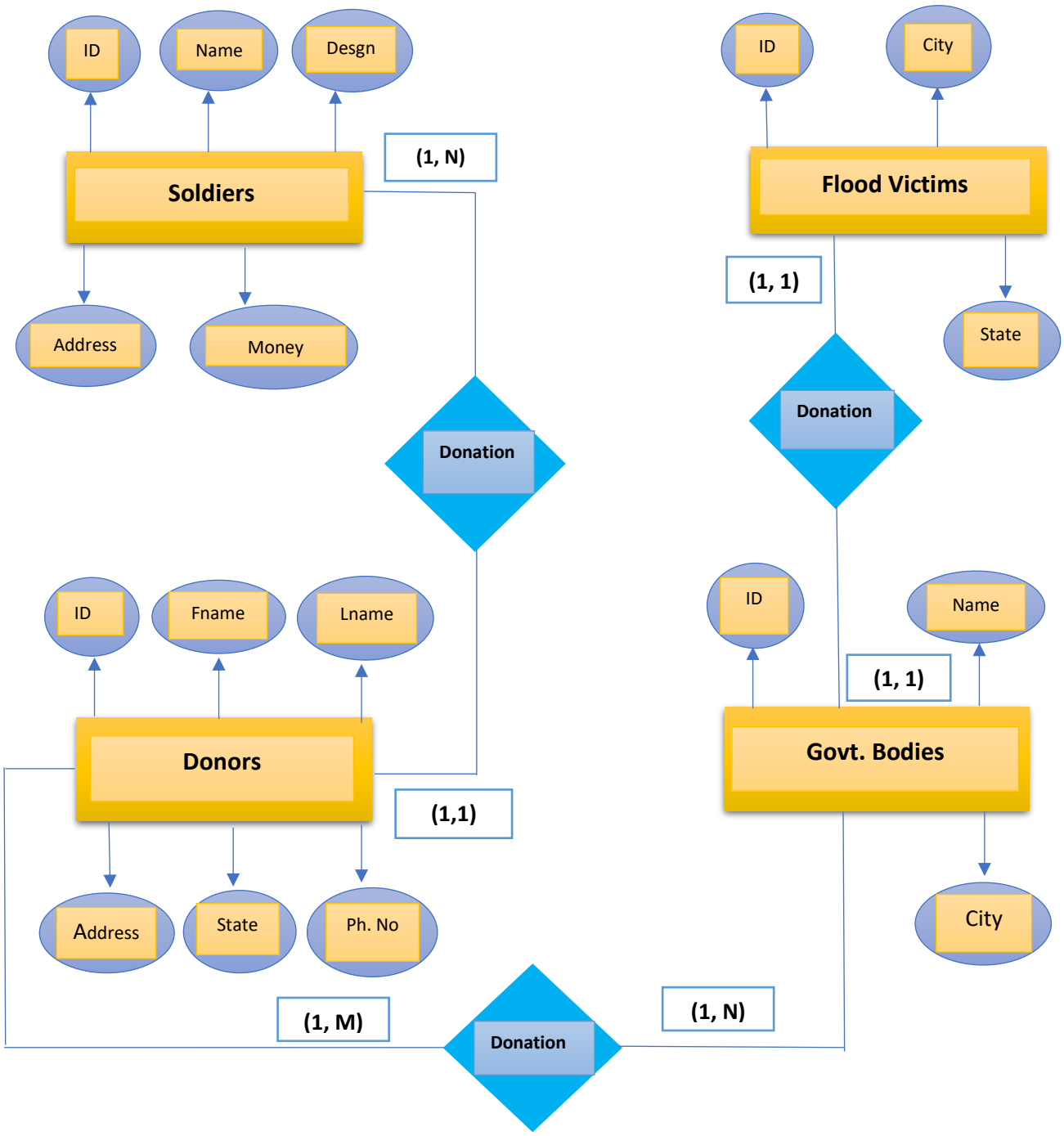
- **Operating System** – Windows 7, 8, 10
- **RAM** – 1 GB or more
- **Processor Chip** – Intel core i3, i5, i7

2.2 Software Requirements:

- **Language** – Python
- **Database** – MySQL
- **Application** – Python IDLE, MySQL Workbench, MySQL Command Client

CHAPTER 3

DATA MODELS AND ER MODEL



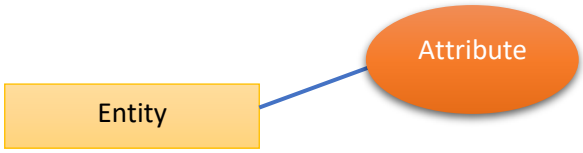
3.1 ENTITY AND ATTRIBUTES

An **Entity** is a real-world object that a user uses to store his/her information in the database. In database administration, only those things about which data will be captured or stored is considered an entity. Some of the examples of an entity are Customer, Doctor, Soldier, Employee etc.

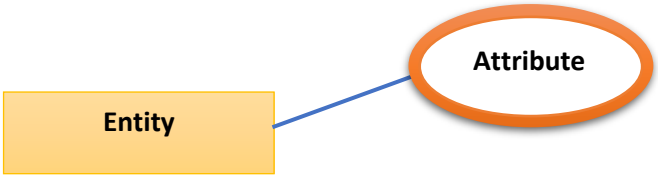
An **Attribute** always contains the data has to be stored inside the database. An entity can have many attributes but all those attributes can belong to only to one entity.For example, a Soldier can contain many attributes such as id, name, Designation, etc., but these attributes can have only one entity.

There are many types of attributes. They are:

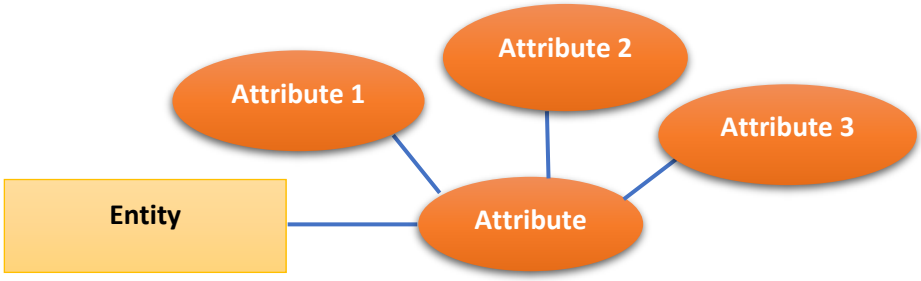
- **Single valued attribute:** An attribute which contains only one single value of the entity is called a single-valued attribute.



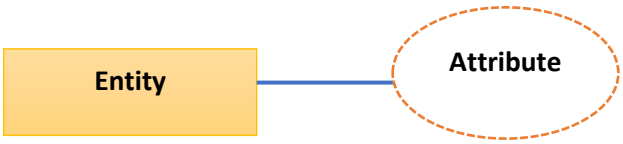
- **Multi-valued attribute**: An Attribute which have multiple values for the same entity is called multi-valued attribute. For example, a solider contains many entities such as id, name, designation etc.



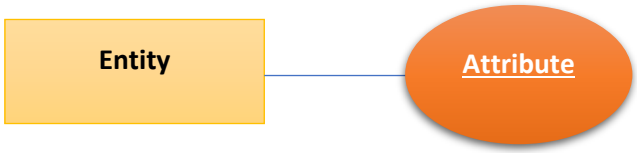
- **Composite attribute**: An attribute that are divided into smaller independent attributes is called composite attribute. For example, an employee contains attributes such as id, name, address, designation etc. Here, the attribute 'address' can contain many attributes such as Street, City, State etc.



- **Derived attribute**: An attribute that can be derived from other attributes is called derived attribute.



- **Key attribute**: An attribute that has an unique value for each entity is called key attribute. The attribute name is always underlined while displaying in ER Model.



The **entities** and **attributes** in the following table are as follows:

Entity:

1) **Soldier**: The table which contains the list of soldiers.
2) **Flood Victims**: Contains the list of organisations funding the flood-affected victims
3) **Govt_bodies**: Contains the list of the relief funds of various states.
4) **Donors**: Contains the list of all donors who donate money either to the soldiers or the organisations which fund the flood-affected victims.

Attributes:

ENTITY	ATTRIBUTES
SOLDIERS	**ID, NAME, ADDRESS, DESIGNATION, TOTAL_COLLECTION**
FLOOD VICTIMS	**ID, CITY, STATE**
DONORS	**ID, FNAME, LNAME, ADDRESS, STATE, PHNO**
GOVT_BODIES	**ID, NAME, CITY**

3.2 KEYS

In MySQL, keys are a set of fields (one or many) that is always unique for a record. The key column can never contain a NULL column. Keys can also be used to create a relation between two tables. The different types of keys present in MySQL are:

- **Primary key:** The column in the table that uniquely defines every row inside a record is called primary key. The value entered inside the primary key cannot be repeated more than once. The primary key column cannot be NULL.

- **Foreign Key:** The column which is present in one table which is used to create the relationship between two tables is called foreign key. The main advantage of the foreign key is that it is always ensures data integrity.

- **Super Key:** A super key is an attribute which is used to uniquely identify a tuple. It contains a set of all the candidate keys.

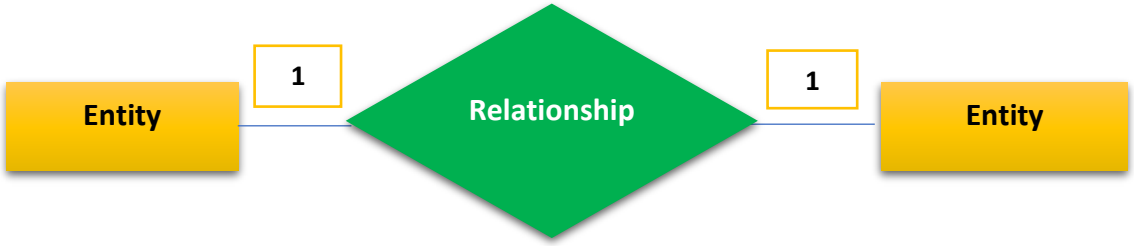
- **Candidate Key:** A key which does not contain any repeated attributes is called candidate key. Every table must contain at least one candidate key. It cannot contain NULL values.

3.3 RELATIONSHIP AND PARTICIPATION

A relationship in MySQL is an association between different entities in an ER model. There are various types of relation in an ER model. They are:

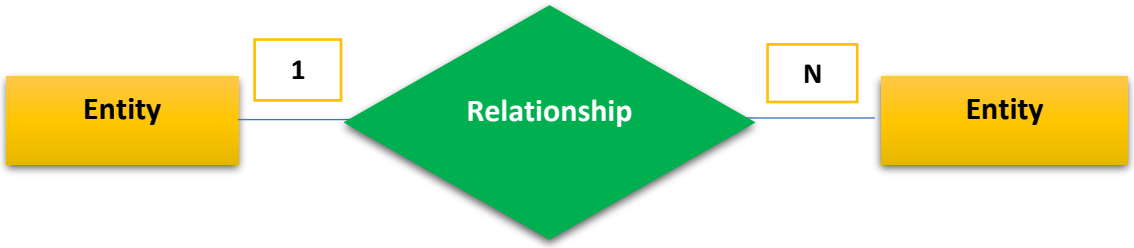
- **1:1 Relationship:**

  In database administration, only those things about which data will be captured or stored is considered an entity.



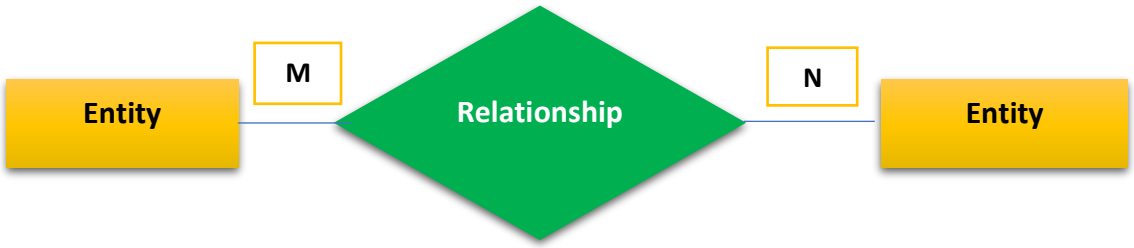
- **1:N Relationship:**

  This is called 'One to Many' relationship. The primary key table contains only one record that relates to none, one, or many records in the related table.



- **M:N relationship:**

  Each record in both tables can relate to any number of records (or no records) in the other table. For instance, a project can be sponsored by many sponsors and a sponsor can sponsor many projects.



3.4 PROBLEM STATEMENT(ASSUMPTIONS)

The assumptions are as follows:

- There are 4 entities. They are: soldiers, Flood-Affected Victims, Donors, and Govt bodies.
- The soldier entity contains ID, Name, Designation, Address, DontionCollected.
- The Flood-Affected Victims entity contains ID, City, State.
- The Donors entity contains ID, FirstName, LastName, Address, State, PhoneNumber.
- The Govt Bodies Entity contains GID, Gname, City, State.
- A donor can donate money to as many soldiers that he/she wants but the soldier can accept donation only once from each donor.
- A donor can donate money to several government bodies and the government bodies can also accept donation from many donors.
- A government body can donate money only to its respective city/state it governs and the city can also accept donation only from its respective city's relief fund.

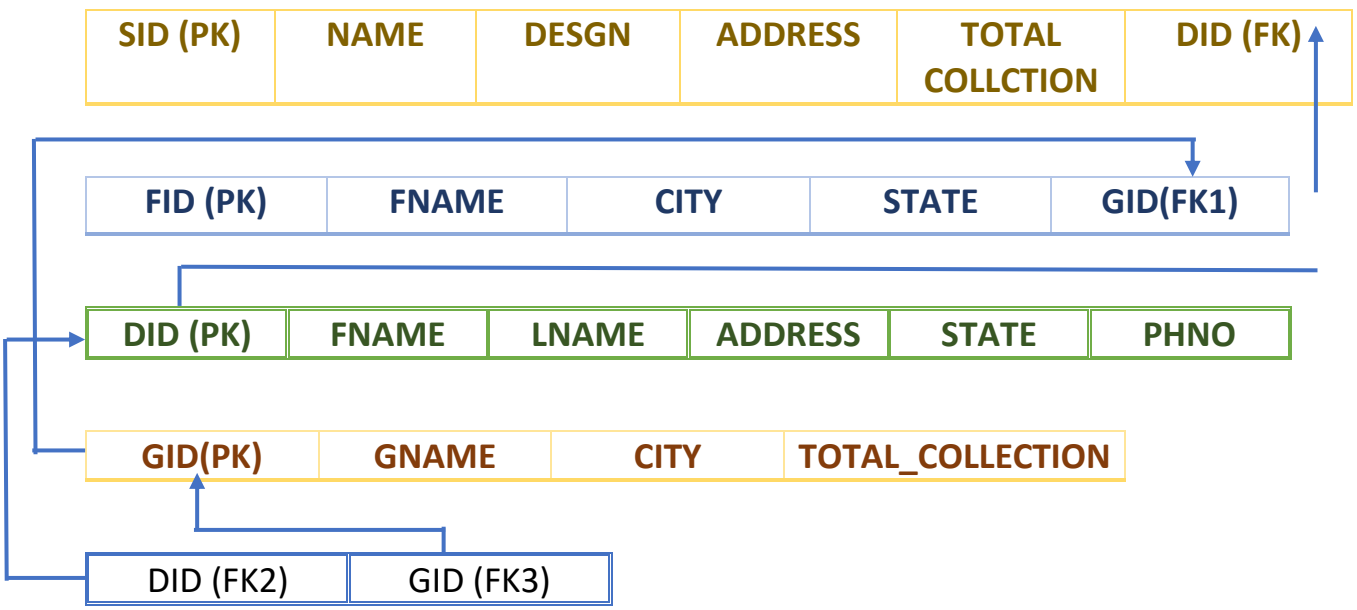
CHAPTER 4

RELATIONAL MODEL CONCEPTS AND SCHEMA

The various relational model concepts in MySQL are:

- **Tables** – In relational data model, relations are saved in the format of Tables. This format stores the relation among entities. A table has rows and columns, where rows represents records and columns represent the attributes.
- **Tuple** – A single row of a table, which contains a single record for that relation is called a tuple.
- **Relation instance** – A finite set of tuples in the relational database system represents relation instance. Relation instances do not have duplicate tuples.
- **Relation schema** – A relation schema describes the relation name (table name), attributes, and their names.
- **Relation key** – Each row has one or more attributes, known as relation key, which can identify the row in the relation (table) uniquely.
- **Attribute domain** – Every attribute has some pre-defined value scope, known as attribute domain

Schema:



CHAPTER 5

SQL

SQL, as the name suggests, stands for Structured Query Language. This language is basically used to connect with a database. It is the standard language for relational database management systems. This language is use to preform various operations such as creating a table to store data, inserting values into the table, and performing various operations with the table created such as updating the table, retrieving the values from the table, deleting the values from a table etc. Some systems where SQL is used are: Oracle, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system.

There are many types of query languages in MYSQL. They are:

- **Data Definition Language (DDL):** This type of language is used for the description of the database and is also used to create and modify the structure of the database.
  Some of the examples of DDL commands are:
  1) **CREATE** – It is used to create the database and records of the database.
     Syntax – CREATE TABLE TABLENAME (COLUMN 1 DT 1, COLUMN 2, DT 2…);

  2) **DROP** – It is used to delete or remove the existing tables or records in the database.
     Syntax – DROP TABLE TABLENAME;

  3) **ALTER**-It is used to alter or modify the structure of the database.
     Syntax – ALTER TABLE TABLENAME ACTION_TYPE ACTION;

  4) **TRUNCATE**–It is used to remove all the values present in the table. The difference between truncate and drop is that while truncate command is executed, the description of the table is not deleted but when DROP command is executed, the values along with the description of the table will be deleted.
     Syntax – TRUNCATE TABLE TABLENAME;

- **Data Manipulation Language (DML):** These commands are particularly used for the manipulation of the data present in the database. The commands are:

  1) INSERT – This command is used to insert values inside the table.
     Syntax – INSERT INTO TABLENAME VALUES (VALUE1, VALUE2,….);

  2) UPDATE – It is used to update the values present in a specific table.
     Syntax – UPDATE TABLENAME SET PARAMETER WHERE CONDITION;

3) **DELETE –** It is used to delete the values present in the existing table.
   Syntax – DELETE FROM TABLENAME WHERE CONDITION;

- **Data Control Language (DCL):** These commands are used to control the rights and permissions in the database systems.

  1) **GRANT –** This command is used to give access privileges to the user controlling the database.
     Syntax – GRANT USER USERNAME ALL ON DATABASE. *;

  2) **REVOKE –** It is used to withdraw the user's access privileges granted to the user using GRANT command.
     Syntax – REVOKE ALL PRIVILEGES, GRANT OPTION FROM USER;

- **Data Query Language (DQL) –** There is only one DQL command that is used to retrieve data from a database. The command used for data query is SELECT.
  SELECT – Use to display data present in a table.
  Syntax – SELECT * FROM TABLENAME;

- **Transaction Control Language (TCL):** It is used to relate with issues regarding transactions in a database.
  The TCL commands are as follows:
  1) **COMMIT** – It is use to save all the work done in the database and confirm all changes.
     Syntax – COMMIT;

  2) **ROLLBACK –** It is used to undo all the changes made in the database.
     Syntax – ROLLBACK;

.

CHAPTER 6

PYTHON FEATURES

There are many features of Python. They are:

6.1 SIMPLE AND EASY TO LEARN:

Python is a very simple language which almost feels like reading English. It is very easy to learn as well. In this language, the concentration is more on the problem rather than the language itself.

6.2 FREE AND OPEN SOURCE:

Python is an open-source software. The user can see the code displayed, access the contents inside it, manipulate the code, create his/her own code etc. Engineers who code always can manipulate the code inside it to create a better Python.

6.3 PORTABLE:

Python is capable of being used on various platforms due to its open-source nature. It can be used on various systems without any changes in it's software. Python can be used on Linux, Windows, Solaris etc.

6.4 OBJECT-ORIENTED:

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

6.5 HIGHLY DYNAMIC:

Python is one of the most dynamic languages available in the industry today. The need of of specifying the type of the variable is not required, thus increasing the efficiency of execution.

CHAPTER 7

TKINTER WIDGETS

Widgets are basically a medium through which operations in GUI take place. They act as a building block in a GUI application. It consists of various types such as Label, Entry, Button, Messagebox, Combobox etc. The widgets which have been used in the project are:

7.1 Label Widget:

Label widget is one of the most basic widgets in a GUI. The main function of a label widget is to display a string variable. It is used to provide am message to the user and other GUI widgets.

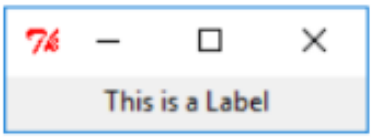
Syntax – Label_1=Label (master, options…)

Options include text to be entered, background colour, foreground colour, font etc.

For example,

Label_1=Label(root, master="This is a label")

The output would be something like this:



7.2 Entry Widget:

Entry widget is a widget which gives the permission to the user to enter a text or a number and provide the input to the GUI. It supports both strings and numbers. IF the string typed by the user is longer than the entry box available, the user will still be able to type his text. The user needs to scroll down to view his whole string entered.

Syntax – entry_1=Entry(master,options….)

The options include textvariable etc.

For example,

w=Label(root,text="Username");

e=Entry(root);

The output would be something like this:

7.3 Button Widget:

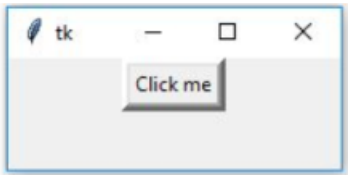
A button is a standard widget in a GUI. It contains almost the same parameters as Label and Entry widget. The speciality of the button widget is that it contains a parameter called 'command'. The function of the command parameter is when a function is assigned to the command parameter, when the user clicks the button in the GUI, the button widget calls the function assigned to the command parameter.

Syntax – button_1=Button(master,options,command)

For example,

Button_1=Button(root,text="Click me")

The output would be something like this:



7.4 Messagebox Widget:

Messagebox widget is a standard widget of Tkinter which is used to display the message boxes once it is called.

Syntax – Messagbox.function_name(title,message)

The function_name consists of various functions. They are:

- showinfo() - used to display basic information.
- Showerror() – used to display errors.
- askquestion() – used to ask a question (basically Yes or No)
- askyesnocancel() – used to ask a question including yes,no and cancel.

For example,

Messagebox.showinfo(“Information”,”information”);

The output would be something like this:

7.5 Message widget:

The message widget is used to display short messages and texts inside a GUI. The difference between Message and Label is flexibility. Message is more flexible in displaying texts when compared to labels.

Syntax – Message(master,options…)

For example,

A=StringVar()

Message(root,text=A)

A.set(“Hey! How are you doing?)

The output would be something like this:

7.6 Combobox widget:

Tkinter Combobox is the drop-down list for the user to choose from. It is a combination of Entry and drop-down menu. When you click the arrow on the left side, you will see a drop-down menu showing all the choices, and if you click on one, it will replace the current Entry contents. We need to import a library named ttk for the combobox to work inside the GUI.

Syntax – from tkinter import ttk

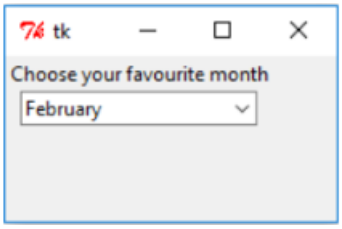
ttk.Combobox(master,values=[…])

For example,

From tkinter import ttk

ttk.Combobox(root,values=[“January”,”February”,”March”,”April”])

The output of the example would be something like this:

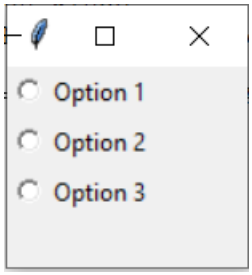


7.7 Radiobutton Widget:

Radiobuttons are standard buttons used in tkinter to allow a user to do a particular selection from a list of selections. It generally implements only one selection.

Syntax – Checkbutton(master,options…)

A sample of a radiobutton widget used in a GUI:



CHAPTER 8

IMPLEMENTATION

FETCHING DONOR DETAILS:

The code below is the first GUI created which allows the donor donating the money to start his process of donation. He/she is asked to fill his/her details i.e., Name, Address, State, E-mail, and Phone Number. The submit button mentioned allows the user to access the second window.

```
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
from PIL import Image,ImageTk
import mysql.connector
root=Tk()
root.geometry('600x600')
root.configure(background="thistle1");
root.title("Relief Funds")
canvas = Canvas(root,width=600,height=600)
canvas.pack()
pilImage = Image.open("background.png")
image = ImageTk.PhotoImage(pilImage)
img= canvas.create_image(0,0,anchor=NW,image=image)
fn=StringVar()
ln=StringVar()
ad=StringVar()
st=StringVar()
rad=IntVar()
em=StringVar()
ph=IntVar()
sld1=0
sld2=0
a=0
b=0
sel=0
sel1=0
minidb=mysql.connector.connect(
     host="localhost",
     user="root",
     passwd="(Hello123)",
     database="Mini"
)

minic=minidb.cursor()
```

```
label_1=Label(root,text="WELCOME USER",width=40,fg="white",bg="blue",font=("arial",12,"bold")).place(x=90,y=20)
label_1=Label(root,text="WE THANK YOU FOR TAKING THIS INIATIVE",width=40,fg="white",bg="blue",font=("arial",12,"bold")).place(x=90,y=45)
label_2=Label(root,text="Enter your details:",width=15,fg="white",bg="blue",font=("arial",12,"bold")).place(x=10,y=80)
label_3=Label(root,text="First Name:",width=15,fg="white",bg="blue",font=("arial",11,"bold")).place(x=10,y=120)
entry_1=Entry(root,textvar=fn).place(x=175,y=120)
label_4=Label(root,text="Last Name:",width=15,fg="white",bg="blue",font=("arial",11,"bold")).place(x=10,y=170)
entry_2=Entry(root,textvar=ln).place(x=175,y=170)

label_5=Label(root,text="Address:",width=15,fg="white",bg="blue",font=("arial",11,"bold")).place(x=10,y=220)
entry_3=Entry(root,textvar=ad).place(x=175,y=220)
label_6=Label(root,text="State:",width=15,fg="white",bg="blue",font=("arial",11,"bold")).place(x=10,y=270)
entry_4=Entry(root,textvar=st).place(x=175,y=270)
label_5=Label(root,text="E-Mail:",width=15,fg="white",bg="blue",font=("arial",11,"bold")).place(x=10,y=320)
entry_5=Entry(root,textvar=em).place(x=175,y=320)
label_6=Label(root,text="Phone No:",width=15,fg="white",bg="blue",font=("arial",11,"bold")).place(x=10,y=370)
entry_6=Entry(root,textvar=ph).place(x=175,y=370)
print(rad)
button_1=Button(root, text='Submit',width=20,bg='brown',fg='white',font=("arial",12,"bold"),command=second_win).place(x=170,y=470)

root.mainloop()
```

INSERTION AND SELECTION:

Here, once SUBMIT button is clicked by the user, if any of the entries is not filled, it pops up a message which says," Please enter your details.". Once all details are entered, the cursor of MySQL i.e., "minic" will execute an SQL statement to insert all the values into the backend MySQL table "DONOR". Once this is done, a second GUI pops up which gives the user three options i.e., Donation for Soldiers being option 1 ,Donation for flood-Affected victims being option 2, and Admin being option 3. User can select an any option he/she wants to donate to. Radio buttons are used for this purpose.

```
def second_win():

    fn1=fn.get();
    ln1=ln.get();
    ad1=ad.get();
    st1=st.get();
    em1=em.get()
    ph1=ph.get();

    if fn1=="" or ln1=="" or ad1=="" or st1=="" or em1=="" or em1=="" or ph1==0:
        messagebox.showerror("Error","Please Enter your details")

    else:

        minic.execute("INSERT INTO DONOR (FNAME,LNAME,ADDRRESS,STATE,PHONE_NUMBER) VALUES(%s,%s,%s,%s,%s)",(fn1,ln1,ad1,st1,ph1));
        minidb.commit()

        window=Tk()
        window.geometry('300x300')
        window.configure(background="thistle1")
        window.title("Relief Funds")
        '''canvas1= Canvas(window,width=300,height=300)
        canvas1.pack()
        pilImage1= Image.open("background.png")
        image1= ImageTk.PhotoImage(pilImage1)
        img1= canvas.create_image(0,0,anchor=NW,image=image1)'''
        hie=IntVar(master=window)
        label_7=Label(window,text="Choose An Option:",width=20,bg="magenta",fg="white",font=("arial",12,"bold")).pack()
        button_2=Radiobutton(window, text="Donation for Soldiers",padx=5, variable=hie,value=1).place(x=10,y=70)
        #print(var)
        button_3=Radiobutton(window, text="Donation for Flood-Affected Victims",padx=5,variable=hie,value=2).place(x=10,y=120)
#       print(var1)
        button_4=Radiobutton(window, text="Admin",padx=5,variable=hie,value=3).place(x=10,y=170)
        #print(hie)
        button_5=Button(window, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:third_win(hie)).place(x=30,y=220)
        button_6=Button(window, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window.destroy).place(x=150,y=220)
```

PROCESS AFTER SELECTION:

Once the user selects his/her option, a GUI pops up based on the option selected. If option 1 is selected, a GUI opens which contains a drop-down list of all the soldiers who have sacrificed their life for this nation. The user can select one soldier among the list provided. If option 2 is selected, a drop-down list which contains the list of the relief funds of various states. The user can select one option among the given list of relief funds. Combo box has been used for this purpose.

```
def third_win(hie):
    print("selection:",hie.get())
    global sel
    sel=hie.get()
    if sel==1:
        window2=Tk()
        window2.geometry('550x300')
        window2.configure(background="thistle1")
        window2.title("Relief Funds")
        cb=StringVar(master=window2)
        '''canvas2= Canvas(window2,width=550,height=300)
        canvas2.pack()
        pilImage2= Image.open("background.png")
        image2= ImageTk.PhotoImage(pilImage2)
        img2= canvas.create_image(0,0,anchor=NW,image=image2)'''
        label_8=Label(window2,text="In the dropdown menu below,\n there is a list of soldiers who have sacrificed their life for this nation.\n Choose the name of the s
        cb=ttk.Combobox(window2,values=['Karnail Singh','Ravi Paul','Rakesh Singh','Javra Munda','Naiman Kajur','Janrao','SK Vidyarthi','G Sanokhar','Rajesh KR SIngh','
        cb.place(x=140,y=110)
        cb.config(width=40)
        cb.current(1)
        print(cb.current(),cb.get())
        button_7=Button(window2, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:fourth_win(sel,cb)).place(x=120,y=220)
        button_8=Button(window2, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window2.destroy).place(x=270,y=220)


    elif sel==2:
        window3=Tk()
        window3.geometry('550x300')
        window3.configure(background="thistle1")
        window3.title("Relief Funds")
        label_10=Label(window3,text="In the dropdown menu below,\n are the list of foundations which are helping the flood-affected victims.\n Choose th charitable four
        cb=ttk.Combobox(window3,values=["KARNATAKA'S CHIEF MINISTER'S RELIEF FUND","KERALA'S CHIEF MINISTER'S RELIEF FUND","MAHARASHTRA'S CHIEF MINISTER'S RELIEF FUND",
        cb.place(x=140,y=130)
        cb.config(width=40)
        cb.current(1)
        #print(cb.current(),cb.get())
        button_7=Button(window3, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:fourth_win(sel,cb)).place(x=120,y=220)
        button_8=Button(window3, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window3.destroy).place(x=270,y=220)
```

```
elif sel==3:
    window3=Tk()
    window3.geometry('300x300')
    window3.configure(background="thistle1")
    window3.title("Relief Funds")
    var=IntVar(master=window3)
    label_7=Label(window3,text="Choose An Option:",width=20,bg="thistle1",font=("arial",12,"bold")).pack()
    button_2=Radiobutton(window3, text="Soldiers",padx=5, variable=var,value=1).place(x=10,y=70)
    button_3=Radiobutton(window3, text="Goverment Relief Fund",padx=5,variable=var,value=2).place(x=10,y=120)
    button_4=Radiobutton(window3, text="Donor",padx=5,variable=var,value=3).place(x=10,y=170)
    button_5=Button(window3, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:eighth_win(var)).place(x=30,y=220)
    button_6=Button(window3, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window3.destroy).place(x=150,y=220)
```

PROCESS AFTER SELECTION – PART 2:

The variables, **sel** (which contains the selected value of second GUI), **cb** (contains the selected value of the soldier from the drop-down menu) are passed to the next function via "**lambda**" keyword which is used to pass variables from one function to another.

- If option 1 was selected before, after selecting the soldier's name, a GUI opens which firstly show the Soldier's name, Bank Name, and Account Number. Here "minic" retrieves the data from the backend, MySQL which contains a table called "SOLDIERS" which contains the bank name and account number of each and every soldier. Secondly, the user is asked for his/her bank details which includes Name, email, the amount to be donated, the date of transaction, and Bank name. The user has to select a bank name form a list of bank names provided. If there is any field which is not filled, a message box pops up here to enter the details.
- If Option 2 is selected, the name of the government and the bank name and account number is displayed. The cursor, "minic", serves the same purpose here as well for extracting the bank name and account number of the government's relief fund's

account. The user is asked for the same details as above. If there is any field which is not filled, a message box pops up here as well.

```
def fourth_win(sel,cb):
    global sld1
    global a
    global b
    window4=Tk()
    window4.geometry('600x600')
    window4.configure(background="thistle1")
    window4.title("Relief Funds")
    nam=StringVar(master=window4)
    amt=IntVar(master=window4)
    acc=StringVar(master=window4)
    date=StringVar(master=window4)
    if sel==1:
        print(cb.get())
        #window4=Tk()
        '''window4.geometry('600x600')
        window4.configure(background="thistle1")
        window4.title("Relief Funds") '''
        sld1=cb.get()
        print(sld1)
        label_1=Label(window4,text="These are the account details of the soldier selected:",width=40,fg="white",bg="magenta",font=("arial",12,"bold")).pack()
        label_12=Label(window4,text="Soldier Name:",bg='purple',fg='white',font=("arial",10,"bold")).place(x=185,y=30)
        Message(window4,text=sld1,width=100,justify=CENTER,bg='blue',fg='white',font=("airal",10,"bold")).place(x=295,y=30)
        label_13=Label(window4,text="Bank Name:",bg='purple',fg='white',font=("arial",10,"bold")).place(x=185,y=70)
        minic.execute("SELECT BANK_NAME FROM soldiers WHERE NAME=(%s)",(sld1,))
        a=minic.fetchall()
        #print(b)
        Message(window4,text=a,width=200,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).place(x=280,y=70)
        label_14=Label(window4,text="Account Number:",bg='purple',fg='white',font=("arial",10,"bold")).place(x=185,y=110)
        minic.execute("SELECT ACC_NO FROM SOLDIERS WHERE NAME=(%s)",(sld1,))
        b=minic.fetchall()
        Message(window4,text=b,width=200,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).place(x=310,y=110)
        label_1=Label(window4,text="Enter your bank details:",width=30,fg="white",bg="magenta",font=("arial",12,"bold")).place(x=130,y=150)
        label_2=Label(window4,text="Name:",width=20,font=("arial",12,"bold")).place(x=110,y=210)
        entry_1=Entry(window4,textvar=nam).place(x=250,y=210)
        label_3=Label(window4,text="Email ID:",width=20,font=("arial",12,"bold")).place(x=110,y=250)
        entry_2=Entry(window4,textvar=acc).place(x=250,y=250)
        label_4=Label(window4,text="Amount:",width=20,font=("arial",12,"bold")).place(x=110,y=290)
        entry_3=Entry(window4,textvar=amt).place(x=250,y=290)
        label_5=Label(window4,text="Date:",width=20,font=("arial",12,"bold")).place(x=110,y=330)
        entry_2=Entry(window4,textvar=date).place(x=250,y=330)
        label_6=Label(window4,text="Select the name of the bank:",width=30,font=("arial",12,"bold")).place(x=150,y=370)
        cb=ttk.Combobox(window4,width=40,values=['Axis Bank','Bank Of Baroda','State Bank Of India','Canara Bank','Indian Overseas Bank','Federal Bank','ICICI Bank'])
        cb.place(x=170,y=420)
        button_1=Button(window4, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:fifth_win(sel,cb,nam,acc,amt)).place(x=140,y=520)
        button_2=Button(window4, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window4.destroy).place(x=330,y=520)
```

```
elif sel==2:
    '''window4=Tk()
    window4.geometry('600x600')
    window4.configure(background="thistle1")
    window4.title("Relief Funds") '''
    label_1=Label(window4,text="The account dtetials are listed below:",width=40,fg="white",bg="magenta",font=("arial",12,"bold")).pack()
    sld1=cb.get()
    print(sld1)
    label_12=Label(window4,text="Governnment Name:",bg='purple',fg='white',font=("arial",10,"bold")).place(x=115,y=30)
    Message(window4,text=sld1,width=500,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).place(x=265,y=30)
    label_13=Label(window4,text="Bank Name:",bg='purple',fg='white',font=("arial",10,"bold")).place(x=115,y=70)
    minic.execute("SELECT BANK_NAME FROM govt_body WHERE GNAME=(%s)",(sld1,))
    a=minic.fetchall()
    print(a)
    #print(b)
    Message(window4,text=a,width=200,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).place(x=265,y=70)
    label_14=Label(window4,text="Account Number:",bg='purple',fg='white',font=("arial",10,"bold")).place(x=115,y=110)
    minic.execute("SELECT ACC_NO FROM govt_body WHERE GNAME=(%s)",(sld1,))
    b=minic.fetchall()
    Message(window4,text=b,width=200,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).place(x=265,y=110)
    label_1=Label(window4,text="Enter your bank details:",width=30,fg="white",bg="magenta",font=("arial",12,"bold")).place(x=130,y=150)
    label_2=Label(window4,text="Name:",width=20,font=("arial",12,"bold")).place(x=110,y=210)
    entry_1=Entry(window4,textvar=nam).place(x=250,y=210)
    label_3=Label(window4,text="Email ID:",width=20,font=("arial",12,"bold")).place(x=110,y=250)
    entry_2=Entry(window4,textvar=acc).place(x=250,y=250)
    label_4=Label(window4,text="Amount:",width=20,font=("arial",12,"bold")).place(x=110,y=290)
    entry_3=Entry(window4,textvar=amt).place(x=250,y=290)
    label_5=Label(window4,text="Date:",width=20,font=("arial",12,"bold")).place(x=110,y=330)
    entry_2=Entry(window4,textvar=date).place(x=250,y=330)
    print(amt)
    label_6=Label(window4,text="Select the name of the bank:",width=30,font=("arial",12,"bold")).place(x=150,y=370)
    cb=ttk.Combobox(window4,width=40,values=['Axis Bank','Bank Of Baroda','State Bank Of India','Canara Bank','Indian Overseas Bank','Federal Bank','ICICI Bank'])
    cb.place(x=170,y=420)
    button_1=Button(window4, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:fifth_win(sel,cb,nam,acc,amt)).place(x=140,y=520)
    button_2=Button(window4, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window4.destroy).place(x=330,y=520)
```

PROCESS AFTER ENTERING BANK DETAILS:

The entry variables **nam** (contains the name of donor), **acc** (contains the email), **amt** (contains the amount to be donated) are passed to the next function along with **sel** and **cb** via **lambda** keyword.

- The name,email, and amount will be stored inside a table in the backend, MySQL, called bank_details_soldiers.
- If Option 1 is continuing, the user will be asked to enter his username and password.

- Once the user clicks submit, the username and password will be stored into the same table, bank_details_soldiers.
- After storage is complete, a GUI will pop up which shows the final display of details which contains the name of soldier his bank name, his account number and that total amount to be paid.
- Once the user checks his/her details and clicks the proceed, the amount entered by the user will be stored in the soldier's table in MySQL in a column called TOTAL_COLLETION.
- After storage is done, the message box pops up which displays a message showing that the amount has been successfully transferred to the soldier's account.

```
def fifth_win(sel,cb,nam,acc,amt):
    print(nam.get())
    print(amt.get())
    print(acc.get())
    if nam.get()=="" or acc.get()=="" or amt.get()==0:
        messagebox.showerror("Error","Please Enter the details")
    else:
        window5=Tk()
        window5.geometry('450x330')
        window5.title("Relief Funds")
        window5.configure(background='thistle1')
        us=StringVar(master=window5)
        pas=StringVar(master=window5)
        if sel==1:
            minic.execute("INSERT INTO bank_details_soldier (Name,Email,Amount) VALUES(%s,%s,%s)",(nam,acc,amt));
            minidb.commit();
            '''window5=Tk()
            window5.geometry('450x330')
            window5.title("Relief Funds")
            window5.configure(background='thistle1')'''
            label_1=Label(window5,text="The bank selected is:",width=30,bg='magenta',fg='white',font=("arial",12,"bold")).pack()
            sel1=cb.get()
            Message(window5,text=sel1,width=200,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).pack()
            label_1=Label(window5,text="Username:",width=15,font=("arial",12,"bold")).place(x=40,y=80)
            entry_1=Entry(window5,textvar=us).place(x=180,y=80)
            label_2=Label(window5,text="Password:",width=15,font=("arial",12,"bold")).place(x=40,y=130)
            entry_2=Entry(window5,textvar=pas,show="*").place(x=180,y=130)
            button_5=Button(window5, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:sixth_win(sel,cb,us,pas,amt)).place(x=70,y=220)
            button_6=Button(window5, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window5.destroy).place(x=230,y=220)
```

```
def sixth_win(sel,cb,us,pas,amt):
    amt1=amt.get()
    print(us.get())
    print(pas.get())
    if sel==1:
        minic.execute("INSERT INTO bank_details_soldier(username,password) VALUES(%s,%s)",(us,pas));
        minidb.commit();
        window6=Tk()
        window6.geometry('450x450')
        window6.title("Relief Funds")
        window6.configure(background='thistle1')
        label_1=Label(window6,text="Final display of details:",width=30,bg='magenta',fg='white',font=("arial",12,"bold")).pack()
        label_2=Label(window6,text="Soldier Name:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=60)
        Message(window6,text=sld1,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=59)
        label_3=Label(window6,text="Bank Name:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=100)
        Message(window6,text=a,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=99)
        label_4=Label(window6,text="Account NO:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=140)
        Message(window6,text=b,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=139)
        label_2=Label(window6,text="Amount:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=180)
        Message(window6,text=amt1,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=179)
        button_1=Button(window6, text='Proceed',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:seventh_win(sel,cb,amt)).place(x=50,y=280)
        button_2=Button(window6, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window6.destroy).place(x=190,y=280)
```

```
def seventh_win(sel,cb,amt):
    amt1=amt.get()
    if sel==1:
        minic.execute("UPDATE soldiers SET TOTAL_COLLECTION=(%s) WHERE NAME=(%s)",(amt1,sld1))
        minidb.commit()
        messagebox.showinfo("Success","Amount has been transferred to the soldier's account!")
```

- If Option 2 is continuing, the user will be asked again to enter his username and password.
- Once the user clicks submit, the username and password will be stored into the same table, bank_details_fav.
- After storage is complete, a GUI will pop up which shows the final display of details which contains the name of the relief fund selected, its bank name, account number and that total amount to be paid.
- Once the user checks his/her details and clicks the proceed, the amount entered by the user will be stored in the govt_body's table in MySQL in a column called TOTAL_COLLETION.

```
elif sel==2:
    minic.execute("INSERT INTO bank_details_fav (Name,Email,Amount) VALUES(%s,%s,%s)",(nam,acc,amt));
    minidb.commit();
    '''window5=Tk()
    window5.geometry('450x330')
    window5.title("Relief Funds")
    window5.configure(background='thistle1')'''
    label_1=Label(window5,text="The bank selected is:",width=30,bg='magenta',fg='white',font=("arial",12,"bold")).pack()
    sel1=cb.get()
    #print(sel)
    Message(window5,text=sel1,width=200,justify=CENTER,bg='blue',fg='white',font=("arial",10,"bold")).pack()
    label_1=Label(window5,text="Username:",width=15,font=("arial",12,"bold")).place(x=40,y=80)
    entry_1=Entry(window5,textvar=us).place(x=180,y=80)
    label_2=Label(window5,text="Password:",width=15,font=("arial",12,"bold")).place(x=40,y=130)
    entry_2=Entry(window5,textvar=pas,show="*").place(x=180,y=130)
    button_5=Button(window5, text='Submit',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:sixth_win(sel,cb,us,pas,amt)).place(x=70,y=220)
    button_6=Button(window5, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window5.destroy).place(x=230,y=220)
```

```
elif sel==2:
    #minic.execute("INSERT INTO bank_details_fav(username,password) VALUES(%s,%s)",(us,pas));
    #minidb.commit();
    window6=Tk()
    window6.geometry('550x400')
    window6.title("Relief Funds")
    window6.configure(background='thistle1')
    label_1=Label(window6,text="Final display of details:",width=30,bg='magenta',fg='white',font=("arial",12,"bold")).pack()
    label_2=Label(window6,text="Fund Name:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=60)
    Message(window6,text=sld1,width=400,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=59)
    label_3=Label(window6,text="Bank Name:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=100)
    Message(window6,text=a,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=99)
    label_4=Label(window6,text="Account NO:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=140)
    Message(window6,text=b,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=139)
    label_2=Label(window6,text="Amount:",width=15,bg='purple',fg='white',font=("arial",11,"bold")).place(x=20,y=180)
    Message(window6,textvariable=amt1,width=250,justify=CENTER,bg='blue',fg='white',font=("arial",11,"bold")).place(x=170,y=179)
    button_1=Button(window6, text='Proceed',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=lambda:seventh_win(sel,cb,amt)).place(x=100,y=280)
    button_2=Button(window6, text='Cancel',width=10,bg='brown',fg='white',font=("arial",12,"bold"),command=window6.destroy).place(x=240,y=280)
```

```
elif sel==2:
    minic.execute("UPDATE govt_body SET TOTAL_COLLECTION=(%s) WHERE GNAME=(%s)",(amt1,sld1))
    minidb.commit()
    messagebox.showinfo("Success","Amount has been transferred to the Government's's Relief Fund!")
```

CHAPTER 10

CONCLUSION

Therefore, this mini project, i.e., Project on Relief Funds, done using Python's library, Tkinter and the back-end database, MySQL, is performing every task assigned, successfully. From this project, I have been able to gain knowledge about a lot of things. Firstly, I have learnt about the python's library which is used for the creation of a GUI, i.e., Tkinter. It has been very helpful in creating the GUI and the widgets inside it are very helpful is gaining access of the user's data. I have also updated my knowledge on MySQL, which is performing the role of storage of data. I have also learnt about the flow of data i.e., how the data of the user flows from the GUI to the database using the 'MySQL. Connector' option. Therefore, the project has been very useful in learning new things about various features of Python and MySQL.

CHAPTER 11

BIBLIOGRAPHY

The references used for this mini project are:

- www.geeksforgeeks.com
- Let us Python – Yeshawant Kanetkar
- Programming Knowledge – Youtube channel