CHHATTISGARH PUBLIC SCHOOL

Computer Science Project on
GUI Graph Generator
Session-2019-2020

Submitted by

Raghav Pandit

Under the Guidance of

Mr. Sashwat Sharma

Certificate

This is to certify that Raghav Pandit of class 12 has prepared the project on "GUI Graph Generator".The project is the result of his efforts & endeavors. The project is found worthy of acceptance as final project report for the subject Computer Science of class 12. He has prepared the project under my guidance.

(Mr. Sashwat Sharma)
Computer Science
Chhattisgarh Public School

Declaration

I hrerby declare that the project work entitled "Graph Generator" is prepared by me.
All the coding are result of my personal effort.

Raghav Pandit
Class 12 Science

Acknowledgment

I would like to express my special thanks and gratitude to my Computer Science educator Mr. Sashwat Sharma Sir ,who gave me the opportunity to do this wonderful project.

Raghav Pandit
Class 12 science

Contents

Python Libraries used

1. Tkinter
2. Pyplot from Matplotlib
3. Pillow
4. Collections
5. CSV

Working Description

This program is designed to provide a user friendly Graphical user interface for the ease of use.

This Program can generate three type of graphs

1. Bar Graph
2. Pi Chart
3. Line Chart

Coading

```
#Project on GUI graph generator
#This can be used to generate graphs and charts with the help of graphical user interface,
#so it is easily usable to any person
#made by :- Raghav Pandit

import tkinter as tk
from tkinter import messagebox
from tkinter import colorchooser
from PIL import ImageTk,Image
import matplotlib.pyplot as pl
from tkinter.filedialog import askopenfile
import csv
from collections import Counter
pl.style.use("fivethirtyeight")
HEIGHT = 600
WIDTH = 800
lineColor =("adf","black")
markerColor = ("adf","black")

isBar = False
isLine = True



root = tk.Tk()
root.geometry(f"{WIDTH}x{HEIGHT}")
root.title("Graph Generator")

####################__FUNCTIONS_#######################
#Contains all the function required in programm

def reverse(ls):
    return [ele for ele in reversed(ls)]

def
lineChart(xValues,yValues,xLabel,yLabel,ls,lw,color,marker,markersize,markercolor,t
itle):
    xValues=eval(xValues)
    yValues = eval(yValues)
```

```
pl.plot(xValues,yValues,ls=ls,linewidth=lw,color=color,marker=marker,markersize=
markersize,markerfacecolor=markercolor)
    pl.title(title)
    pl.xlabel(xLabel)
    pl.ylabel(yLabel)
    #pl.grid(True)
    print(pl.style.available)

    pl.show()

def barChart(xLabel,yLabel,yValues,xValues,title,width=.8):


    pl.bar(eval(xValues),eval(yValues),width=width)

    pl.xlabel(xLabel)
    pl.ylabel(yLabel)
    pl.title(title)
    pl.show()


def piChart(values,labels,title):
    values = eval(values)
    labels = eval(labels)
    pl.pie(values,labels=labels)
    pl.title(title)
    pl.plot()
    pl.show()



def getMarkerColor():
    global markerColorLabel
    global markerColor
    markerColor = colorchooser.askcolor()
    markerColorLabel.config(bg=markerColor[1])

def getLineColor():
    global colorLabel
    global lineColor
    lineColor = colorchooser.askcolor()
    colorLabel.config(bg=lineColor[1])

def barFrameRaise():
    mainFrameBar.tkraise()
```

```
    global isBar , isLine
    isBar = True
    isLine = False


def lineFrameRaise():
    mainFrameLine.tkraise()
    global isBar , isLine
    isBar = False
    isLine = True


def piFrameRaise():
    mainFramePi.tkraise()
    global isBar , isLine
    isBar = False
    isLine = False


def generate():

    if isLine:

lineChart(xValueLine.get('1.0',tk.END),yValueLine.get('1.0',tk.END),xLabelValueLin
e.get(),yLabelValueLine.get(),ls=lineStyle.get(),lw=lineWidth.get(),color=lineColor[1
],marker=markerStyle.get(),markersize=markerSize.get(),markercolor=markerColor[
1],title=title.get())
    elif isBar:

barChart(xLabelValueBar.get(),yLabelValueBar.get(),yValueBar.get('1.0',tk.END),xV
alueBar.get('1.0',tk.END),barTitle.get(),barWidth.get())
    else:
        piChart(piValues.get('1.0',tk.END),piLabels.get('1.0',tk.END),piTitle.get())

def about():
    messagebox.showinfo("About Creator","This program is created by Raghav Pandit
of class 12 Science")
man='''
Inserting the value:-
To insert value seperate them by commas ',' and use inverted commas for strings but
not for integers
Example :- X-Values = 'val1','val2','val3'
           Y-Values = 10,20,30 '''
def manual():
    messagebox.showinfo("User_Manual",man)
```

```
####################__XXXXXXXXX_#######################


####################__IMAGES__#######################
#Contains All images in need in the programm

#Background image
bgImage = ImageTk.PhotoImage(Image.open(r"Images\background.jpg"))

#Image of line icon
lineImage = ImageTk.PhotoImage(Image.open(r"Images\icons8-line-chart-64.png"))

#Image of bar chart
barImage = ImageTk.PhotoImage(Image.open(r"Images\icons8-bar-chart-
96.png").resize((69,69),Image.ANTIALIAS))

#Image for icon
iconImage = ImageTk.PhotoImage(Image.open(r"Images\appicon.png"))
#Set icon of application
root.iconphoto(False,iconImage)

#Image for heading icon
headingImage =
ImageTk.PhotoImage(Image.open(r"Images\HeadingIcon.png").resize((55,55),Image
.ANTIALIAS))

#image of pi chart icon
piImage = ImageTk.PhotoImage(Image.open(r"Images\icons8-doughnut-chart-
256.png").resize((64,64),Image.ANTIALIAS))

####################__XXXXXX__#######################


####################__WIDGETS__#######################
#Contains all the widgets

#Menu

menu = tk.Menu(root)
helpMenu = tk.Menu(root)
aboutMenu = tk.Menu()

menu.add_cascade(label="Help",menu=helpMenu)
```

```
helpMenu.add_command(label="About",command=about)
helpMenu.add_command(label="User-Manual",command=manual)
root.config(menu=menu)

#Setting background image
imageLabel = tk.Label(root,image=bgImage)
imageLabel.pack()

#Setting Top Frame with Heading
topFrame = tk.Frame(root)
topFrame.place(relx=.005,rely = .005 , relwidth = .985 , relheight = .09)

headingLabel = tk.Label(topFrame,text="Graph
Generator",font=("Caslon",35,"bold"),fg="#26cdeb")
headingLabel.pack()

#icon of the app
mainIconLabel = tk.Label(topFrame,image=headingImage)
mainIconLabel.place(relx=.18,rely=0)

#Side Frame which will contain main navigation of programm
sideFrame = tk.Frame(root)
sideFrame.place(relx=.005,rely = .1 , relwidth = .19 , relheight = .84)


#Side Button for line chart
tk.Button(sideFrame , text="Line Chart" , font = ("Caslon",21,"bold") ,
bg="#e8e8e8",pady=61,relief="raised",command=lineFrameRaise).place(relx=0,rely
=0,relwidth=1,relheight =.3335)


#Side button for bar chart
tk.Button(sideFrame , text="Bar Chart" , font = ("Caslon",21,"bold") ,
bg="#e8e8e8",pady=61,command=barFrameRaise).place(relx=0,rely=.33,relwidth=1
,relheight =.3335)

#Side button for pi chart
tk.Button(sideFrame , text="Pi Chart" , font = ("Caslon",21,"bold") ,
bg="#e8e8e8",pady=61,command=piFrameRaise).place(relx=0,rely=.66,relwidth=1,r
elheight =.3445)


#Generate button
tk.Button(root,text="Generate",font=("Caslon",10,"bold"),command =
```

```
generate).place(relx=.903,rely=.95)

####################__MainPiChart__#####################

#Main frame for pi chart
mainFramePi = tk.Frame(root)
mainFramePi.place(relx=.2,rely=.1 , relwidth = .79 , relheight = .84)

tk.Label(mainFramePi,text="Pi
Chart",font=("Caslon",28,"bold")).place(rely=.03,relx=.13)

piImageLabel = tk.Label(mainFramePi,image=piImage)
piImageLabel.place(relx=.01,rely=.01)

#X-Value and it's text area
tk.Label(mainFramePi,text="Values:-",font=("Caslon",18)).place(relx=.01,rely=.3)
piValues = tk.Text(mainFramePi)
piValues.place(relx=.01,rely = .38,relheight = .20,relwidth=.48)


#Y-Value and it's entry
tk.Label(mainFramePi,text="Labels",font=("Caslon",18)).place(relx=.5,rely=.3)
piLabels = tk.Text(mainFramePi)
piLabels.place(relx=.5,rely=.38,relwidth=.48,relheight = .20)

#Title of bar graph
piTitle = tk.StringVar()
tk.Label(mainFramePi,text="Title",font=("Caslon",20)).place(relx=.20,rely=.88)
tk.Entry(mainFramePi,textvariable = piTitle,font = ("Caslon",15)).place(relx = .
30,rely=.89)


####################__XXXXXXXXXXXX__#####################



####################__MainBarChart__#####################

#Main frame for bar chart
mainFrameBar = tk.Frame(root)
mainFrameBar.place(relx=.2,rely=.1 , relwidth = .79 , relheight = .84)

tk.Label(mainFrameBar,text="Bar
Chart",font=("Caslon",28,"bold")).place(rely=.03,relx=.13)
```

```
barImageLabel = tk.Label(mainFrameBar,image=barImage)
barImageLabel.place(relx=.01,rely=.01)

#X-Label and it's Entry
xLabelValueBar = tk.StringVar()
tk.Label(mainFrameBar,text="x-label",font=("Caslon",18)).place(relx=.01,rely=.2)
tk.Entry(mainFrameBar,textvariable = xLabelValueBar,font =
("Caslon",15)).place(relx = .12,rely=.21)


#Y-Label and it's Entry
yLabelValueBar = tk.StringVar()
tk.Label(mainFrameBar,text="y-label",font=("Caslon",18)).place(relx=.5,rely=.2)
tk.Entry(mainFrameBar,textvariable = yLabelValueBar,font =
("Caslon",15)).place(relx = .63,rely=.21)

#X-Value and it's text area
tk.Label(mainFrameBar,text="x-
values:-",font=("Caslon",18)).place(relx=.01,rely=.3)
xValueBar = tk.Text(mainFrameBar)
xValueBar.place(relx=.01,rely = .38,relheight = .20,relwidth=.48)


#Y-Value and it's entry
tk.Label(mainFrameBar,text="y-values:-",font=("Caslon",18)).place(relx=.5,rely=.3)
yValueBar = tk.Text(mainFrameBar)
yValueBar.place(relx=.5,rely=.38,relwidth=.48,relheight = .20)


#To set bar width
tk.Label(mainFrameBar,text="Bar
Width",font=("Caslon",18)).place(relx=.5,rely=.59)
barWidth = tk.DoubleVar()
barWidth.set(.8)
tk.Entry(mainFrameBar,font=("Caslon",18),textvariable=barWidth,).place(relx=.70,re
ly=.59,relwidth=.15)


#Title of bar graph
barTitle = tk.StringVar()
tk.Label(mainFrameBar,text="Title",font=("Caslon",20)).place(relx=.20,rely=.88)
tk.Entry(mainFrameBar,textvariable = barTitle,font = ("Caslon",15)).place(relx = .
30,rely=.89)

######################__XXXXXXXXXXXXX__#######################
```

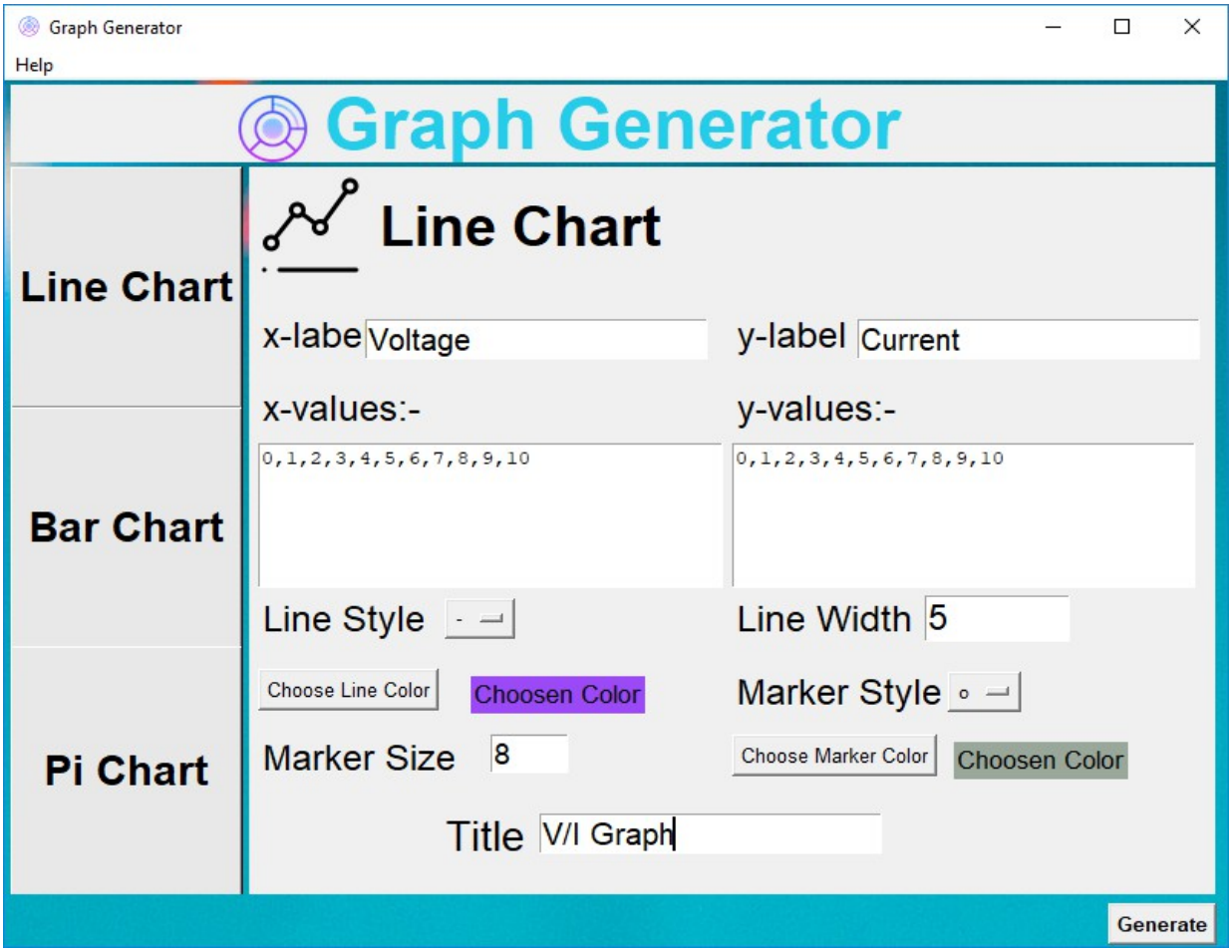
```
####################__MainLineChart__#######################

#Main frame for line chart
mainFrameLine = tk.Frame(root)
mainFrameLine.place(relx=.2,rely=.1 , relwidth = .79 , relheight = .84)

#Top label of the line chart frame
tk.Label(mainFrameLine,text="Line
Chart",font=("Caslon",28,"bold")).place(rely=.03,relx=.13)

#Top icon of the linechart frame
tk.Label(mainFrameLine,image=lineImage).place(relx=.01,rely=.01)


#X-Label and it's Entry
xLabelValueLine = tk.StringVar()
tk.Label(mainFrameLine,text="x-label",font=("Caslon",18)).place(relx=.01,rely=.2)
tk.Entry(mainFrameLine,textvariable = xLabelValueLine,font =
("Caslon",15)).place(relx = .12,rely=.21)


#Y-Label and it's Entry
yLabelValueLine = tk.StringVar()
tk.Label(mainFrameLine,text="y-label",font=("Caslon",18)).place(relx=.5,rely=.2)
tk.Entry(mainFrameLine,textvariable = yLabelValueLine,font =
("Caslon",15)).place(relx = .63,rely=.21)

#X-Value and it's text area
tk.Label(mainFrameLine,text="x-
values:-",font=("Caslon",18)).place(relx=.01,rely=.3)
xValueLine = tk.Text(mainFrameLine)
xValueLine.place(relx=.01,rely = .38,relheight = .20,relwidth=.48)

#Y-Value and it's entry
tk.Label(mainFrameLine,text="y-
values:-",font=("Caslon",18)).place(relx=.5,rely=.3)
yValueLine = tk.Text(mainFrameLine)
yValueLine.place(relx=.5,rely=.38,relwidth=.48,relheight = .20)

#To set line Style
lineStyle = tk.StringVar()
tk.OptionMenu(mainFrameLine,lineStyle,"-","--","-.",":").place(relx=.20,rely=.59)
```

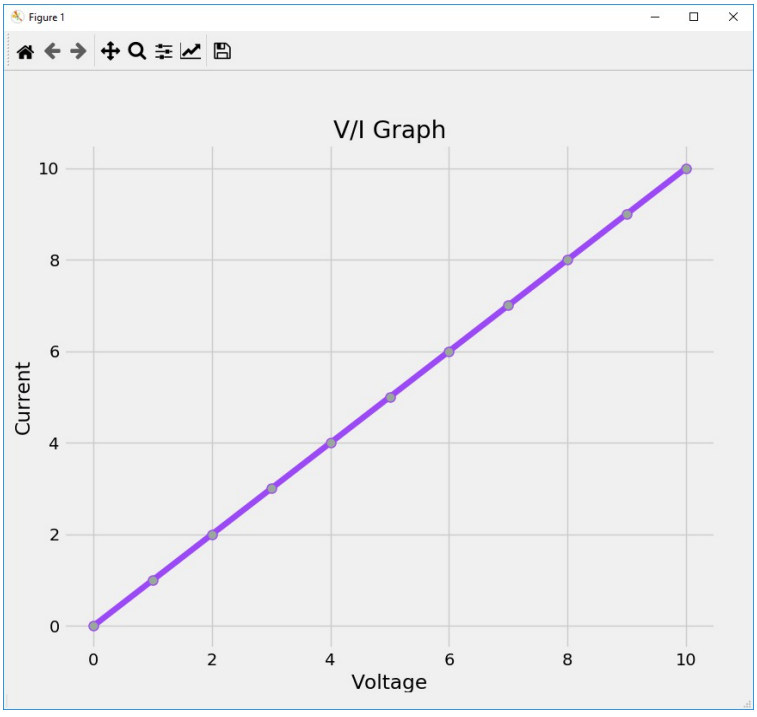
```
lineStyle.set("-")
tk.Label(mainFrameLine,text="Line
Style",font=("Caslon",18)).place(relx=.01,rely=.59)


#To set line width
tk.Label(mainFrameLine,text="Line
Width",font=("Caslon",18)).place(relx=.5,rely=.59)
lineWidth = tk.IntVar()
lineWidth.set(1)
tk.Entry(mainFrameLine,font=("Caslon",18),textvariable=lineWidth,).place(relx=.70,
rely=.59,relwidth=.15)


#Line color
tk.Button(mainFrameLine,text="Choose Line
Color",command=getLineColor,font=("Caslon",10)).place(relx=.01,rely=.69)
colorLabel = tk.Label(mainFrameLine,text="Choosen
Color",font=("Caslon",13),bg="white")
colorLabel.place(relx=.23,rely=.70,relwidth=.18)

#Marker style
tk.Label(mainFrameLine,text="Marker
Style",font=("Caslon",18)).place(relx=.5,rely=.69)
markerStyle = tk.StringVar()
markerStyle.set("None")
tk.OptionMenu(mainFrameLine,markerStyle,"None",".",",","o","+","x","D","d","s","
p","*","h","H","1","2","3","4","v","^","<",">","|").place(relx=.72,rely=.69)


#Maker Size
tk.Label(mainFrameLine,text="Marker
Size",font=("Caslon",18)).place(relx=.01,rely=.78)
markerSize = tk.IntVar()
markerSize.set(1)
tk.Entry(mainFrameLine,font=("Caslon",14),textvariable=markerSize).place(relx=.25
,rely=.78,relwidth=.08)

#Marker color
tk.Button(mainFrameLine,text="Choose Marker
Color",command=getMarkerColor,font=("Caslon",10)).place(relx=.50,rely=.78)
markerColorLabel = tk.Label(mainFrameLine,text="Choosen
Color",font=("Caslon",13),bg="white")
markerColorLabel.place(relx=.73,rely=.79,relwidth=.18)
```

```
#Title of graph
title = tk.StringVar()
tk.Label(mainFrameLine,text="Title",font=("Caslon",20)).place(relx=.20,rely=.88)
tk.Entry(mainFrameLine,textvariable = title,font = ("Caslon",15)).place(relx = .
30,rely=.89)

####################__XXXXXXXXXXXXXXX__#####################

####################__XXXXXXX__#####################
root.mainloop()
```

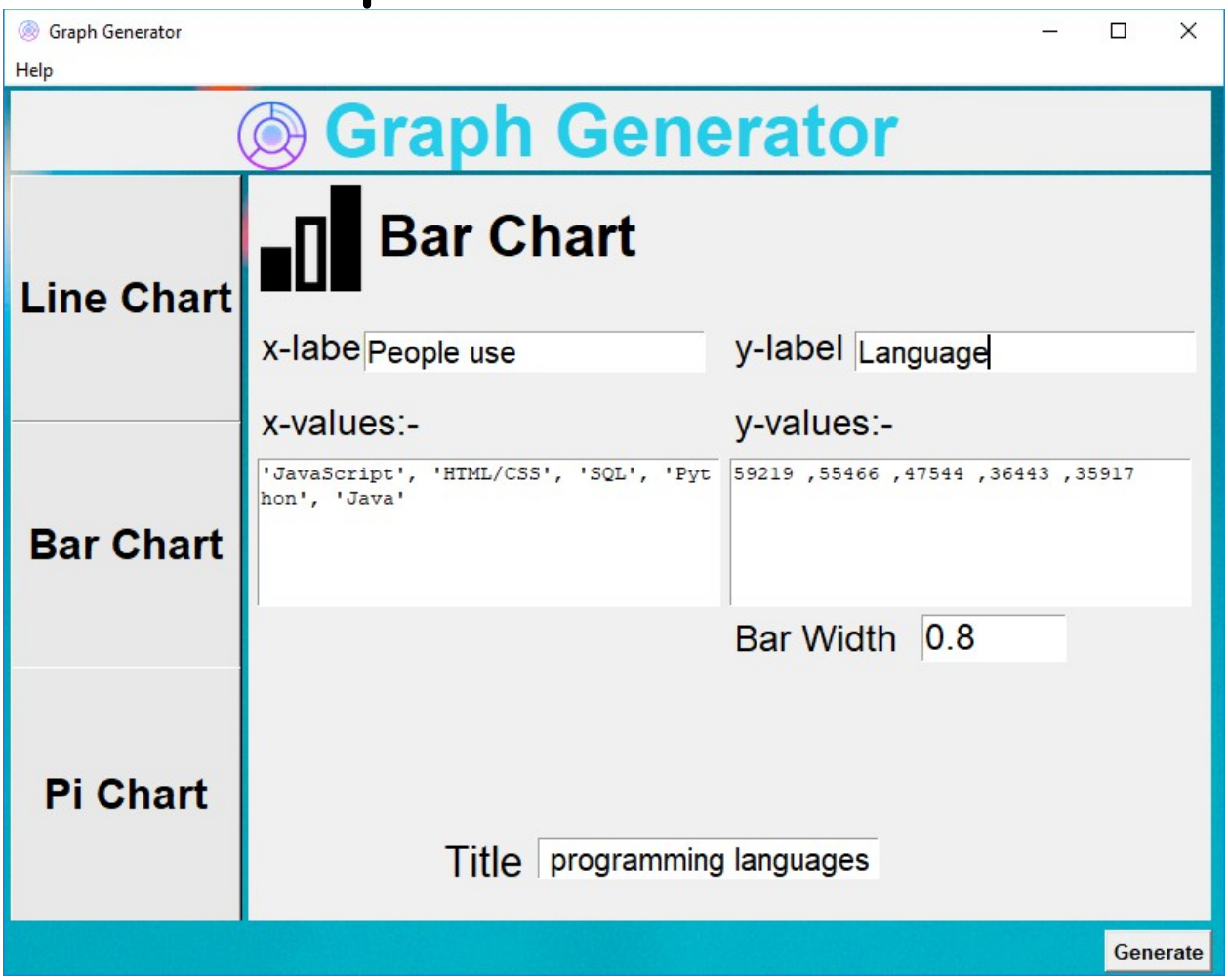
Output

1. Line Chart





2.Bar Graph


Graph Generator
Help
Graph Generator
Line Chart
Bar Chart
Pi Chart
Bar Chart
x-labe
People use
y-label
Language
x-values:-
y-values:-
'JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java'
59219 ,55466 ,47544 ,36443 ,35917
Bar Width
0.8
Title
programming languages
Generate

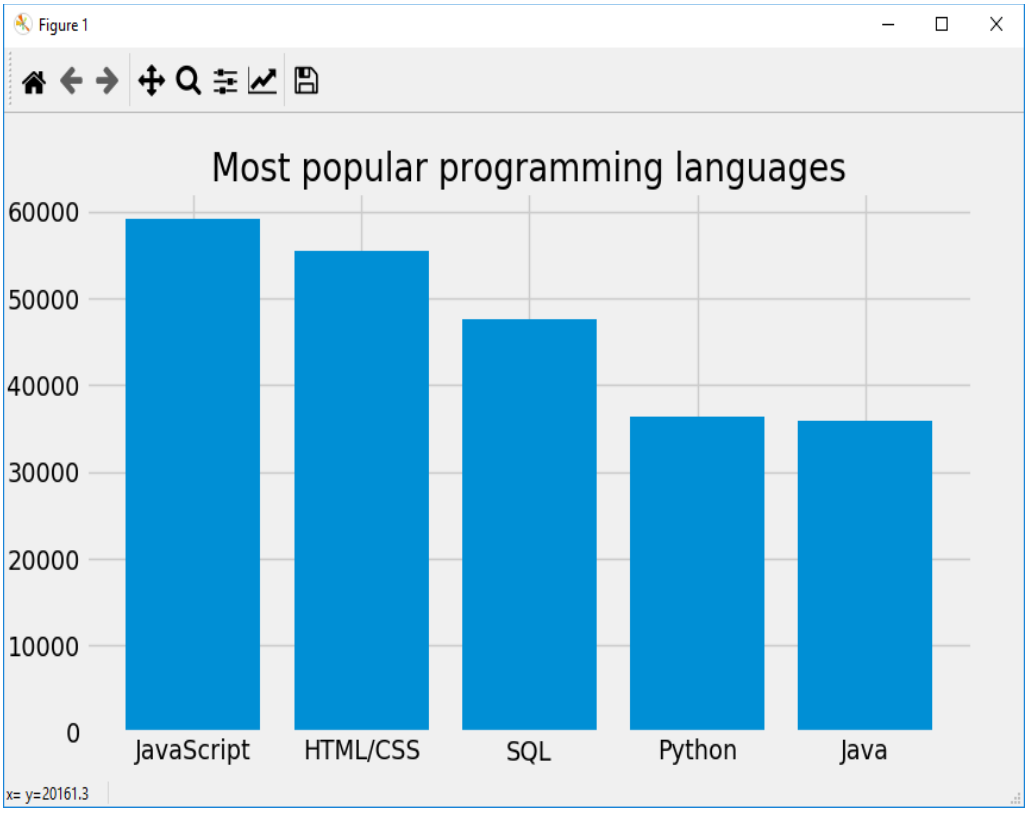
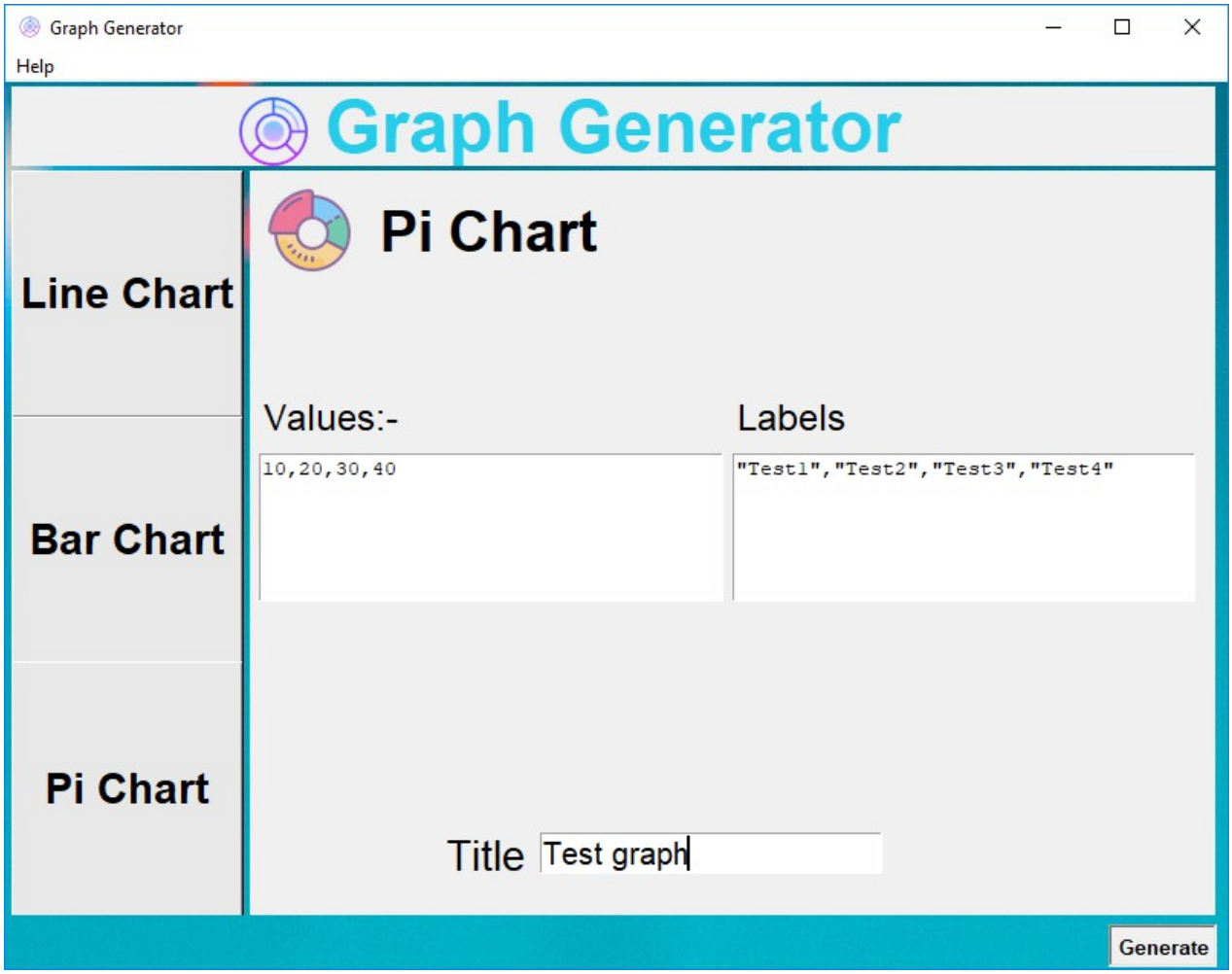


Figure 1
Most popular programming languages
60000
50000
40000
30000
20000
10000
0
JavaScript
HTML/CSS
SQL
Python
Java
x= y=20161.3

3. Pi Chart


Graph Generator
Help
Graph Generator
Line Chart
Bar Chart
Pi Chart
Pi Chart
Values:-
Labels
10,20,30,40
"Test1","Test2","Test3","Test4"
Title
Test graph
Generate

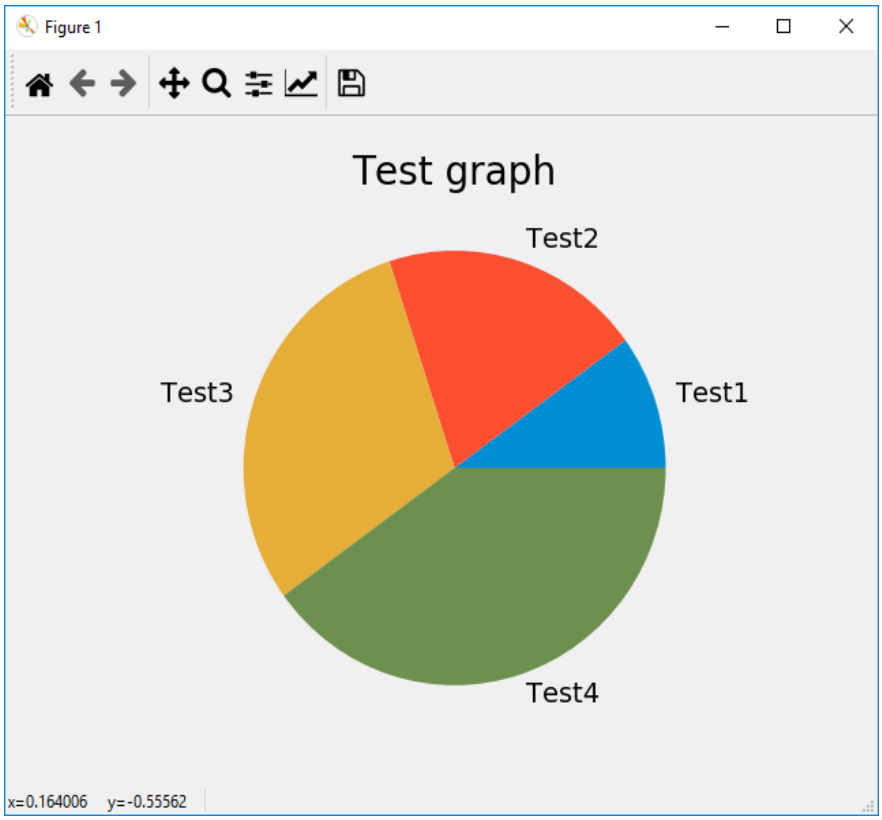


Figure 1
Test graph
Test2
Test3
Test1
Test4
x=0.164006 y=-0.55562

Bibliography

1. www.tutorialspoint.com
2. www.stackoverflow.com
3. www.google.com
4. www.youtube.com