

Objective

The primary objective of this project is to implement what we've learnt throughout our course of Python programming and use that to develop a Graphical User Interface (GUI) for fitness calculator with all the required functionalities. This project also aims at providing a user friendly interface to the users to let them easily use the fitness calculator so that the lack of knowledge in fitness to no hindrance in achieving their fitness goals. Through this project we aim to target the lazy part of the society and let them know that how much they lack in terms of fitness and how much they need to work upon their body to stay fit and healthy while following their strict corporate lifestyle. This project will also help in spreading awareness regarding fitness and people around will be well aware of different criteria, that are include in the field of fitness.

Fitness calculators the are the easiest way of tracking one's fitness level and this is what this project aims for. There are different modules that we've provided on our interface such as Body Mass Index Calculator, VO2 Max Calculator, Body Fat Percentage Calculator, Calorie Calculator, Basal Metabolic Rate Calculator and a Report generator that further includes a few things like Hemoglobin, Blood Pressure, RBC Count, WBC Count, Cholesterol Levels, etc. On the basis of all these parameters provided in the modules a user can check his/her fitness level. In this way, it will help the user to maintain their diet and stay healthy as well as disease free.

This calculator will also generate the detailed report of the user based on the information entered. S.M.A.R.T is an acronym to remind you how to set a goal that maps out exactly what you need to do. These goals are Specific, Measurable, Attainable, Relevant, and Time bound.

Introduction

The innovations in the field of science and technology has made our lives easy and hence there is lack of physical activity which has made today's generation prone to different kind of health related issues. Therefore, it becomes important to take care of our fitness and for that we need to keep track of our fitness, that's where fitness calculators come into play. A Fitness calculator is used to calculate the fitness levels of an individual, and also suggests methods to improve it. This project is no exception, it has been coded in python and comes with a graphical user interface to facilitate the users. This project has 6 different modules which includes, BMI Calculator, Vo2 Max Calculator, Report Generator, BMR Calculator, BFC Calculator and Calorie Calculator. This project also includes SQL database connectivity that helps the use to store their report and then fetch it later, when required. Below given is the description of each module.

- **BMI Calculator**

The body mass index (BMI) or Quetelet index is a value derived from the mass (weight) and height of an individual. The BMI is defined as the body mass divided by the square of the body height, and is universally expressed in units of kg/m^2 , resulting from mass in kilograms and height in meters. The BMI is an attempt to quantify the amount of tissue mass (muscle, fat, and bone) in an individual, and then categorize that person as underweight, normal weight, overweight, or obese based on that value.

This module helps us calculate the BMI of an individual when we input values (height and weight) in our desired units with the help of two radio buttons one for kilograms and centimeters and other for pounds and feet. There is also a reset and back button provide to navigate through screens.

- **VO2 Max Calculator**

VO2 max, also known as maximal oxygen uptake, is the measurement of the maximum amount of oxygen a person can utilize during intense exercise. It is a common measurement used to establish the aerobic endurance of an athlete prior to or during the course of training.

This module helps in calculating the VO2 max levels of an individual when we input out age, weight, heart rate, and time taken to complete a run of one km. This module also comes with a chart that tells the standard values for different age groups with different categories, so that the users can compare their calculated values and get to know the category they belong to.

- **Body Fat Calculator**

Body fat includes essential body fat and storage body fat. Essential body fat is a base level of fat that is found in most parts of the body. It is necessary fat that maintains life and reproductive functions. The amount of essential fat differs between men and women, and is typically around 2-5% in men, and 10-13% in women. The healthy range of body fat for men is typically defined as 8-19%, while the healthy range for women is 21-33%.

This module lets the user calculate their body fat percentage so that they could work upon improving it. Here we've used the BMI method to calculate the body fat percentage by taking input values of age, weight, height and gender.

- **Basal Metabolic Rate Calculator**

The Basal Metabolic Rate (BMR) Calculator estimates your basal metabolic rate—the amount of energy expended while at rest in a neutrally temperate environment, and in a post-absorptive state (meaning that the digestive system is inactive, which requires about 12 hours of fasting).

This module lets the user calculate their basal metabolic rate based on their age, gender, height and weight. Here we've used the Mifflin - St Jeor equation to estimate BMR.

- **Calorie Calculator**

Calories are units of energy, contained within food, and used by the human body to maintain daily health and life. Calories are associated with energy that is contained in protein, carbohydrates and fat. Within each of these measures, calories will have a given amount of energy available.

This module lets the user estimate the number of calories a person needs to consume each day based on your activity, gender, age, height, weight, etc.

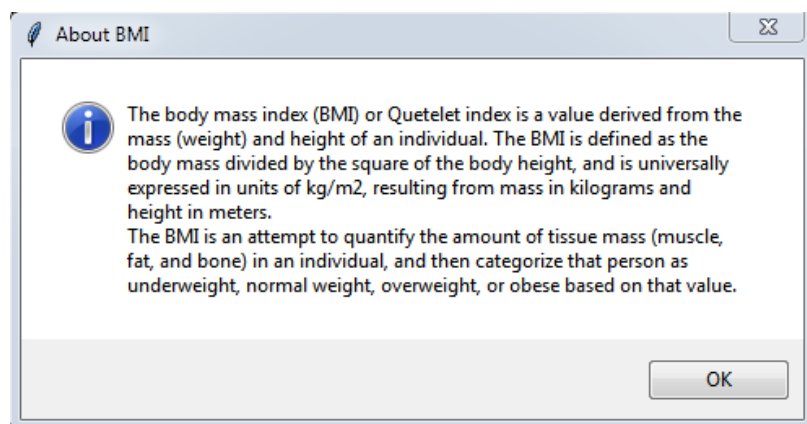
- **Report Generator**

This module tell a person their level of Blood pressure, Hemoglobin, Cholesterol, RBC count, WBC count and Pulse rate. It tells whether your levels of these inputs are low, normal or high. This module takes input values of systolic BP, diastolic BP, cholesterol levels, RBC count, WBC count, Hemoglobin levels and pulse rate.

This module lets the user save their generated report to the database using database connectivity (sqlite3), after saving record the ‘Save to Database’ becomes ‘Saved’ and disabled therefore you can save record once only. Afterwards you can fetch your saved report from the database using a button ‘Fetch Record’. Unsaved reports can’t be fetched. To fetch a record, you’ll have to enter the name and if that name is present the record will be displayed and if not present you’ll be prompted to the main window with a message ‘Record not found’.

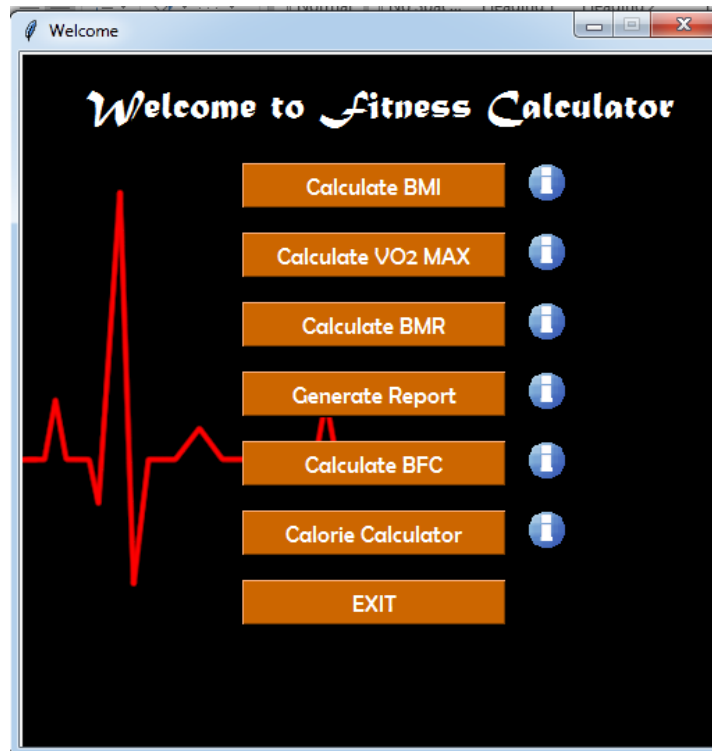
- **The Info Button**

This button is present in the welcome screen which looks a blue ‘I’, it is present next to every module button and on clicking upon it, you’ll get a popup dialog box with the information regarding the module it is placed next to. It is given with a purpose to let the users know the significance of each module that we’ve provided. It looks like this (*after clicking info next to BMI*).

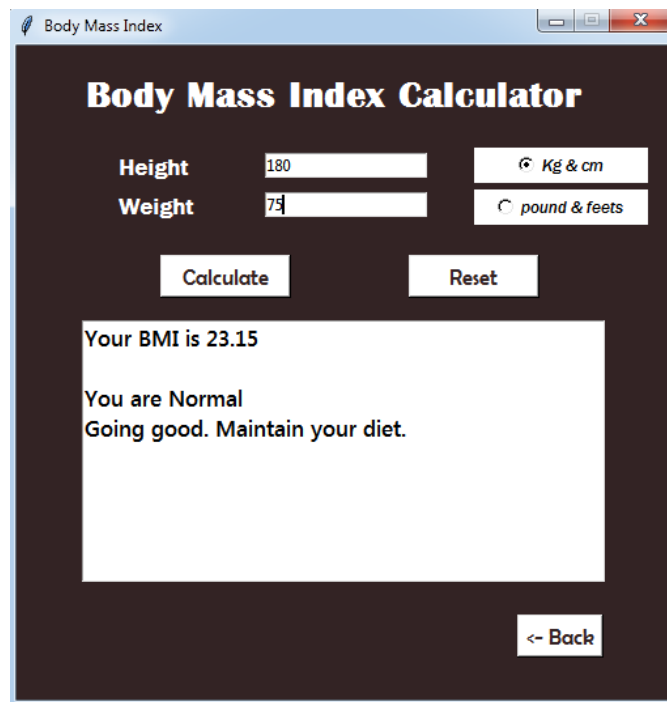


GUI Screenshots:

1.WELCOME SCREEN:



2.BMI CALCULATOR (with input values)



3. VO2 MAX CALCULATOR (with input values)

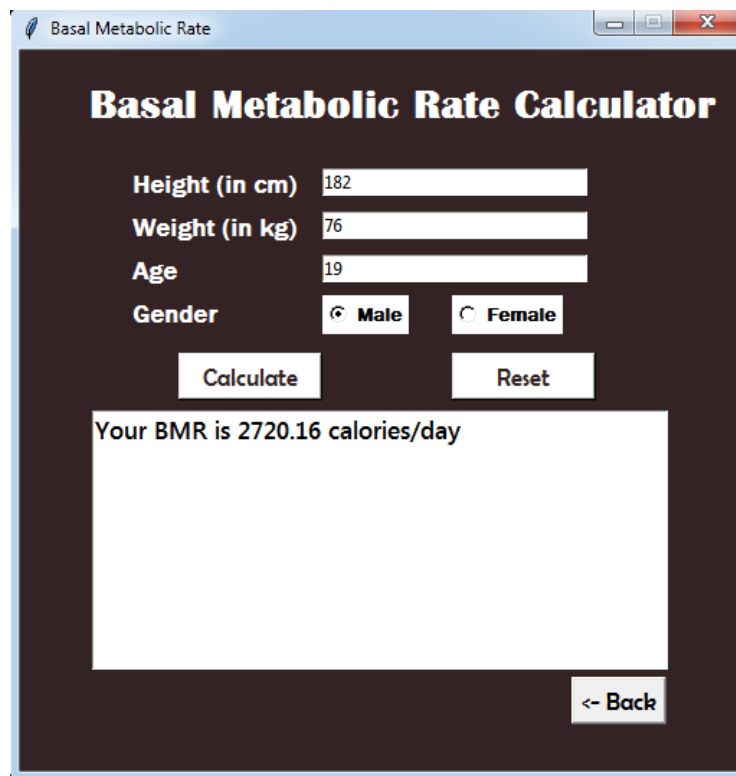
The screenshot shows a web application window titled "VO2". The main heading is "VO2 Max Calculator". Below the heading are five input fields: "Weight(in Kg):" with the value "75", "Age(in Years):" with the value "19", "Heart Rate" with the value "70", "Time(in Min):" with the value "17", and "Gender:" with radio buttons for "Male" (selected) and "Female". Below these fields are two buttons: "Calculate" and "Reset". A large white box displays the result: "Your vo2 is 53.54". At the bottom of the window are two buttons: "click here to view standard values" and "<- Back".

4. STANDARD VALUES (on clicking 'View Standard Values')

The screenshot shows a web application window titled "Vo2 Values". At the top left is a "<- Back" button. The main content is a table with columns for "Age (years)", "Superior", "Excellent", "Good", "Fair", and "Poor". The table is divided into two sections: "Men" and "Women".

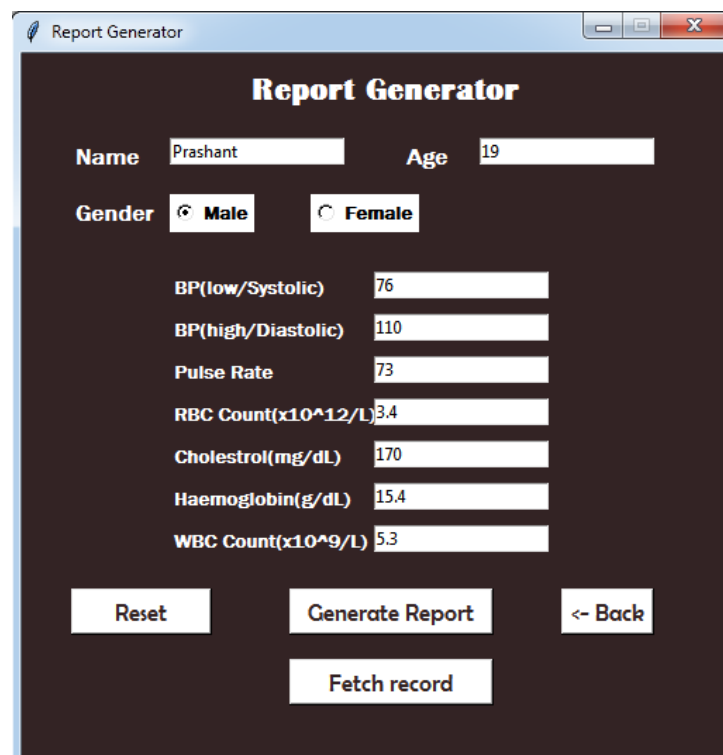
Age (years)	Superior	Excellent	Good	Fair	Poor
Men					
20-29	56+	51-55	46-50	42-45	≤ 41
30-39	54+	48-53	44-47	41-43	≤ 40
40-49	53+	46-52	42-45	38-41	≤ 37
50-59	50+	43-49	38-42	35-37	≤ 34
60-69	46+	39-45	35-38	31-34	≤ 30
70-79	42+	36-41	31-35	28-30	≤ 27
Women					
20-29	50+	44-49	40-43	36-39	≤ 35
30-39	46+	41-45	37-40	34-36	≤ 33
40-49	45+	39-44	35-38	32-34	≤ 31
50-59	35+	31-34	29-30	25-28	≤ 24
60-69	36+	32-35	29-31	26-28	≤ 25
70-79	36+	30-35	27-29	24-26	≤ 23

5. BMR CALCULATOR (with input values)



The screenshot shows a window titled "Basal Metabolic Rate" with a dark blue background. The title "Basal Metabolic Rate Calculator" is centered at the top in white. Below the title, there are four input fields: "Height (in cm)" with the value 182, "Weight (in kg)" with the value 76, "Age" with the value 19, and "Gender" with radio buttons for "Male" (selected) and "Female". Below these fields are two buttons: "Calculate" and "Reset". A large white text box displays the result: "Your BMR is 2720.16 calories/day". At the bottom right, there is a button labeled "<- Back".

6. REPORT GENERATOR (with input values)



The screenshot shows a window titled "Report Generator" with a dark blue background. The title "Report Generator" is centered at the top in white. Below the title, there are four input fields: "Name" with the value Prashant, "Age" with the value 19, "Gender" with radio buttons for "Male" (selected) and "Female", and a list of medical parameters with their values: "BP(low/Systolic)" 76, "BP(high/Diastolic)" 110, "Pulse Rate" 73, "RBC Count($\times 10^{12}/L$)" 3.4, "Cholestrol(mg/dL)" 170, "Haemoglobin(g/dL)" 15.4, and "WBC Count($\times 10^9/L$)" 5.3. Below these fields are three buttons: "Reset", "Generate Report", and "<- Back". At the bottom center, there is a button labeled "Fetch record".

7. GENERATED REPORT (after clicking 'Generate Report')

Report Generator

Generated Report

->Enter name whose record you want to fetch<-

Name	Prashant
Age	19
BP	High BP
Pulse Rate	Average
RBC Count	Normal count
Cholestrol	Normal
Haemoglobin	Low
WBC Count	Normal count

Fetch record Save to database <- Back

Saved

8. FETCH RECORD (on clicking 'Fetch Button' after entering 'Name')

Report Generator

Generated Report

->Enter name whose record you want to fetch<-

Name	Prashant
Age	19
BP	High BP
Pulse Rate	Average
RBC Count	Normal count
Cholestrol	Normal
Haemoglobin	Low
WBC Count	Normal count

Fetch record Save to database <- Back

9. BODY FAT CALCULATOR (with input values)

Body Fat Calculator

Body Fat Calculator

Height (in cm) 182

Weight (in kg) 75

Age 19

Gender ☒ Male ☐ Female

Calculate Reset

Your Bodt Fat Percentage is 15.34

<- Back

10. CALORIE CALCULATOR (with input values)

Body Fat Calculator

Body Fat Calculator

Height (in cm) 182

Weight (in kg) 75

Age 19

Gender ☒ Male ☐ Female

Calculate Reset

Your Bodt Fat Percentage is 15.34

<- Back

Source Code

```
from tkinter import*
import sys
from tkinter import messagebox
import sqlite3

x=sqlite3.connect("fc.db")
c=x.cursor()

class welcome(): #welcome window

    def __init__(self, master):
self.master=master
self.master.geometry("500x500+500+180")
self.master.resizable(0,0)
self.master.title("Welcome")
self.master.configure(background="#e6ffcc")
        photo=PhotoImage(file='hrt.gif')
self.limage=Label(self.master, image=photo,width=500, height=500)
self.limage.image=photo
self.limage.place(x=0, y=0)
self.title=Label(self.master, text="Welcome to Fitness Calculator", font=("Matura MT Script Capitals",
22), bg="black", fg="white").place(x=45, y=19)
        self.bmi=Button(self.master, text="Calculate BMI", bg="#cc6600",fg="white", font=("Berlin Sans
FB",13), command=self.gotobmi, width=20).place(x=160, y=80)#bmibtn
        pic=PhotoImage(file='infos.gif')
self.infobmi=Button(self.master, image=pic, width=27, height=27, bg="black", bd=0,
command=self.bmiinfo)#infobmi
self.infobmi.image=pic
self.infobmi.place(x=365, y=80)
        self.vo2=Button(self.master, text="Calculate VO2 MAX", bg="#cc6600",fg="white",
font=("Berlin Sans FB",13), command=self.gotovo2, width=20).place(x=160, y=130)#vo2btn
self.infovo2=Button(self.master, image=pic, width=27, height=27, bg="black", bd=0,
command=self.vo2info)#infovo2
self.infovo2.image=pic
self.infovo2.place(x=365, y=130)
        self.vo2=Button(self.master, text="Calculate BMR", bg="#cc6600",fg="white", font=("Berlin
Sans FB",13), command=self.gotobmr, width=20).place(x=160, y=180)#bmrbutton
self.inforeport=Button(self.master, image=pic, width=27, height=27, bg="black", bd=0,
command=self.bmrinfo)#infobmr
self.inforeport.image=pic
```

```

self.inforeport.place(x=365, y=180)
self.report=Button(self.master, text="Generate Report", bg="#cc6600",fg="white", font=("Berlin Sans
FB",13), command=self.gotoreport, width=20).place(x=160, y=230)#reportbtn
self.inforeport=Button(self.master, image=pic, width=27, height=27, bg="black", bd=0,
command=self.reportinfo)#inforeport
self.inforeport.image=pic
self.inforeport.place(x=365, y=230)
    self.bmr=Button(self.master, text="Calculate BFC", bg="#cc6600",fg="white", font=("Berlin Sans
FB",13), command=self.gotobfc, width=20).place(x=160, y=280)#bfcbtn
self.infobmr=Button(self.master, image=pic, width=27, height=27, bg="black", bd=0,
command=self.bfcinfo)#infobfc
self.infobmr.image=pic
self.infobmr.place(x=365, y=280)
    self.cal=Button(self.master, text="Calorie Calculator", bg="#cc6600",fg="white", font=("Berlin
Sans FB",13), command=self.gotocalorie, width=20).place(x=160, y=330)#bfcbtn
self.infocal=Button(self.master, image=pic, width=27, height=27, bg="black", bd=0,
command=self.calorieinfo)#infobmi
self.infocal.image=pic
self.infocal.place(x=365, y=330)
self.exit=Button(self.master, text="EXIT", bg="#cc6600",fg="white", font=("Berlin Sans FB",13),
command=self.gotoexit, width=20).place(x=160, y=380)

def bmiinfo(self):
    msg="about bmi "
messagebox.showinfo("About BMI", msg)

def reportinfo(self):
    msg="This module generates a patient report.
messagebox.showinfo("About Report", msg)

def bfcinfo(self):
    msg="This module calculates the body fat percentage of an individual person.
messagebox.showinfo("About BFC", msg)

def calorieinfo(self):
    msg="The Calorie Calculator can be used to estimate the number of calories a person needs to
consume each day. "
messagebox.showinfo("About Calorie Count", msg)

def gotobmi(self):
    root2=Toplevel(self.master)
    bmgui=bmi(root2)

```

```

def gotovo2(self):
    root3=Toplevel(self.master)
    vo2gui=vo2(root3)

def gotobmr(self):
    root4=Toplevel(self.master)
    bmrgui=bmr(root4)

def gotoreport(self):
    root5=Toplevel(self.master)
    reportgui=report(root5)

def gotoexit(self):
    self.master.destroy()

class bmi():    #calculate BMI

    def __init__(self, master):
self.master=master
self.master.geometry("500x500+500+180")
self.height=DoubleVar()
self.weight=DoubleVar()
self.choice=IntVar()
self.master.resizable(0,0)
self.master.title("Body Mass Index")
self.master.configure(background="#332324")
self.title=Label(self.master, text="Body Mass Index Calculator",font=("Britannic Bold",22, "bold"),
fg="white", bg="#332324" ).place(x=50, y=18)
self.lheight=Label(self.master, text="Height ", font=("Franklin Gothic Medium",13,"bold"),
fg="white", bg="#332324")
self.lheight.place(x=75, y=80)
self.eheight=Entry(self.master, textvariable=self.height, width=20, bg="white")
self.eheight.delete(0, 'end')
self.eheight.focus()
self.eheight.place(x=190,y=82)
self.metric=Radiobutton(self.master, text="Kg & cm",font=("Franklin Gothic Medium",10,"italic"),
variable=self.choice,width=15, value=0, bg="white", fg="black")
self.metric.place(x=350, y=78)
    self.us=Radiobutton(self.master, text="pound & feets",font=("Franklin Gothic
Medium",10,"italic"), variable=self.choice,width=15, value=1, bg="white", fg="black")
self.us.place(x=350, y=110)

```

```

self.lweight=Label(self.master, text="Weight ", font=("Franklin Gothic Medium",13, "bold"),
fg="white", bg="#332324").place(x=75, y=110)
self.eweight=Entry(self.master, textvariable=self.weight, width=20, bg="white")
self.eweight.delete(0, 'end')
self.eweight.focus()
self.eweight.place(x=190,y=112)
self.clear=Button(self.master, text="Reset", command=self.clr,font=("Berlin Sans FB",13),
fg="#332324", bg="white", width=10).place(x=300, y=160)
self.calculate=Button(self.master, text="Calculate",font=("Berlin Sans FB",13),
command=self.calculatebmi, fg="#332324", bg="white", width=10).place(x=110,y=160)
self.result=Text(self.master, state=DISABLED, bg="white",font=("Segoe UI Semibold",13))
self.result.place(x=50, y=210, width=400, height=200)
self.back=Button(self.master, text="<- Back",font=("Berlin Sans FB",13), command=self.back,
fg="#332324", bg="white").place(x=383, y=435)

```

```

def calculatebmi(self):    #caluclate button
    ch=self.choice.get()
    rheight=self.height.get()
    rweight=self.weight.get()
    if ch==1:
        rweight*=703
        rheight*=12
        rheight*=rheight
        bmi=rweight/rheight
    elif ch==0:
        rheight=rheight/100
        rheight*=rheight
        bmi=rweight/rheight
    if(bmi<18.5):
        msg="\n\nYou are underweight\nTry to consume more calories than you burn\n"

    elif(bmi>=18.5 and bmi<25):
        msg="\n\nYou are Normal\nGoing good. Maintain your diet."
    elif(bmi>=25 and bmi<30):
        msg="\n\nYou are Overweight\nYou need to cut down your calorie intake and do some physical
activity"
    else:
        msg="\n\nYou are Obese\nTry to loose weight and if you can't consult a doctor "
    self.result.config(state=NORMAL)
    self.result.insert(END, "Your BMI is %.2f " % bmi + msg)
    self.result.see("end")
    self.result.config(state=DISABLED)

```

```
def clr(self):    #reset button
self.eheight.delete(0, 'end')
self.eweight.delete(0, 'end')
self.result.config(state=NORMAL)
self.result.delete('1.0', END)
self.result.config(state=DISABLED)
```

```
def back(self):    #back button
self.master.destroy();
```

```
class report():    #Report Generator
```

```
def __init__(self, master):
self.master=master
self.master.geometry("500x500+500+180")
self.nm=StringVar()
    self.ag=IntVar()
self.gndr=StringVar(value="Male")
    self.bl=IntVar()
    self.bh=IntVar()
    self.pl=IntVar()
    self.rb=DoubleVar()
self.wb=DoubleVar()
    self.cl=IntVar()
    self.hmb=DoubleVar()
self.master.resizable(0,0)
self.master.title("Report Generator")
self.master.configure(background="#332324")
self.title=Label(self.master, text="Report Generator",font=("Britannic Bold",17, "bold"), fg="white",
bg="#332324" ).place(x=160, y=9)
self.lname=Label(self.master, text="Name", font=("Franklin Gothic Medium",12, "bold"), fg="white",
bg="#332324").place(x=35, y=60)
self.ename=Entry(self.master, textvariable=self.nm, width=20, bg="white")
self.ename.delete(0, 'end')
self.ename.focus()
self.ename.place(x=105,y=60)
self.lage=Label(self.master, text="Age", font=("Franklin Gothic Medium",12, "bold"), fg="white",
bg="#332324").place(x=270, y=60)
self.eage=Entry(self.master, textvariable=self.ag, width=20, bg="white")
self.eage.delete(0, 'end')
```

```
self.eage.focus()
self.eage.place(x=325,y=60)
self.gender=Label(self.master, text="Gender", font=("Franklin Gothic Medium",12, "bold"),
fg="white", bg="#332324").place(x=35, y=100)
self.male=Radiobutton(self.master, text="Male", variable=self.gndr, value="Male", font=("Franklin
Gothic Medium",10, "bold"), fg="black", bg="white")
self.male.place(x=105, y=100)
self.female=Radiobutton(self.master, text="Female", variable=self.gndr, value="Female",
font=("Franklin Gothic Medium",10, "bold"), fg="black", bg="white")
self.female.place(x=205, y=100)
self.bplow=Label(self.master, text="BP(low/Systolic)", font=("Franklin Gothic Medium",10, "bold"),
fg="white", bg="#332324").place(x=105, y=155)
self.ebplow=Entry(self.master, textvariable=self.bl, width=20, bg="white")
self.ebplow.delete(0, 'end')
self.ebplow.place(x=250,y=155)
self.lbphigh=Label(self.master, text="BP(high/Diastolic)", font=("Franklin Gothic Medium",10,
"bold"), fg="white", bg="#332324").place(x=105, y=185)
self.ebphigh=Entry(self.master, textvariable=self.bh, width=20, bg="white")
self.ebphigh.delete(0, 'end')
self.ebphigh.place(x=250,y=185)
self.lpulse=Label(self.master, text="Pulse Rate", font=("Franklin Gothic Medium",10, "bold"),
fg="white", bg="#332324").place(x=105, y=215)
self.epulse=Entry(self.master, textvariable=self.pl, width=20, bg="white")
self.epulse.delete(0, 'end')
self.epulse.place(x=250,y=215)
self.lrbcl=Label(self.master, text="RBC Count( $\times 10^{12}/L$ )", font=("Franklin Gothic Medium",10,
"bold"), fg="white", bg="#332324").place(x=105, y=245)
self.erbc=Entry(self.master, textvariable=self.rb, width=20, bg="white")
self.erbc.delete(0, 'end')
self.erbc.place(x=250,y=245)
self.lcholesterol=Label(self.master, text="Cholesterol(mg/dL)", font=("Franklin Gothic Medium",10,
"bold"), fg="white", bg="#332324").place(x=105, y=275)
self.echolesterol=Entry(self.master, textvariable=self.cl, width=20, bg="white")
self.echolesterol.delete(0, 'end')
self.echolesterol.place(x=250,y=275)
    self.lhb=Label(self.master, text="Haemoglobin(g/dL)", font=("Franklin Gothic Medium",10,
"bold"), fg="white", bg="#332324").place(x=105, y=305)
    self.ehb=Entry(self.master, textvariable=self.hmb, width=20, bg="white")
self.ehb.delete(0, 'end')
self.ehb.place(x=250,y=305)
self.lwbc=Label(self.master, text="WBC Count( $\times 10^9/L$ )", font=("Franklin Gothic Medium",10,
"bold"), fg="white", bg="#332324").place(x=105, y=335)
```



```

self.ewbc=Entry(self.master, textvariable=self.wb, width=20, bg="white")
self.ewbc.delete(0, 'end')
self.ewbc.place(x=250,y=335)
self.clear=Button(self.master, text="Reset", command=self.clr, font=("Berlin Sans FB",13),
fg="#332324", bg="white", width=10).place(x=35, y=380)
self.generate=Button(self.master, text="Generate Report",font=("Berlin Sans FB",13),
command=self.generate, fg="#332324", bg="white", width=15).place(x=190,y=380)
self.fetch=Button(self.master, text="Fetch record",font=("Berlin Sans FB",13), command=self.fetch,
fg="#332324", bg="white", width=15).place(x=190,y=430)
self.back=Button(self.master, text="<- Back",font=("Berlin Sans FB",13), command=self.back,
fg="#332324", bg="white").place(x=383, y=380)

def fetch(self):    #fetch record(button to interface)
    self.name=""
    self.age=""
self.bphigh=""
self.pulse=""
    self.rbc=""
    self.wbc=""
self.hb=""
self.cholesterol=""
    root5=Toplevel(self.master)
    generatedgui=reportout(root5, self.name, self.age, self.bphigh, self.pulse, self.rbc, self.cholesterol,
self.hb, self.wbc)

def generate(self):    #generate report button
    if len(self.ebplow.get())==0 or len(self.ebphigh.get())==0 or len(self.echolesterol.get())==0 or
len(self.ewbc.get())==0 or len(self.erbc.get())==0 or len(self.epulse.get())==0 or len(self.ehb.get())==0
or len(self.ename.get())==0 or len(self.eage.get())==0 :
messagebox.showinfo("attention", "you can't leave any field empty")
    root4=Toplevel(self.master)
    reportgui=report(root4)
    oname=self.nm.get()
obplow=self.bl.get()
obphigh=self.bh.get()
    if obphigh<=90 or obplow<=60:
        obphigh="Low BP"
    elif obphigh<120 and obplow<80:
        obphigh="Normal BP"
    elif obphigh>=120 or obplow>=80:
        obphigh="High BP"
    oage=self.ag.get()

```

```
opulse=self.pl.get()
    if opulse<=55:
        opulse="Athletic"
    elif opulse<=61:
        opulse="Excellent"
    elif opulse<=69:
        opulse="Above average"
orbcs=self.rb.get()
    if orbcs<=2.4:
        orbcs="Low count"
    elif orbcs>2.4 and orbcs<=5.6:
        orbcs="Normal count"
    elif orbcs>5.6:
        orbcs="High count"
owbcs=self.wb.get()
    if owbcs<=3.8:
        owbcs="Low count"
    elif owbcs>3.8 and owbcs<=18.6:
        owbcs="Normal count"
    elif owbcs>18.6:
        owbcs="High count"
ogender=self.gndr.get()
ocholesterol=self.cl.get()
ohb=self.hmb.get()
    if ogender=="Male":
        if ohb<=13.5:
            ohb="Low"
        elif ohb >13.5 and ohb<=17.5:
            ohb="Normal"
        elif ohb>17.5:
            ohb="High"
    elif ogender=="Female":
        if ohb<=12.0:
            ohb="Low"
        elif ohb >12.0 and ohb<=15.5:
            ohb="Normal"
        elif ohb>15.5:
            ohb="High"
    if ocholesterol<=200:
        ocholesterol="Normal"
    else:
        ocholesterol="High"
```

```

root=Toplevel(self.master)
generatedgui=reportout(root, oname, oage, obphigh, opulse, orbc, ocholesterol, ohb, owbc)

def clr(self):    #reset button
self.ename.delete(0, 'end')
self.eage.delete(0, 'end')
self.ebplow.delete(0, 'end')
self.ebphigh.delete(0, 'end')
self.epulse.delete(0, 'end')
self.erbc.delete(0, 'end')
self.ewbc.delete(0, 'end')
self.echolesterol.delete(0, 'end')
self.ehb.delete(0, 'end')

def back(self):    #back button
self.master.destroy();

class reportout(): #Fetch Record

    def __init__(self, master,name, age, bphigh, pulse, rbc, cholesterol, hb, wbc):
self.master=master
self.master.geometry("500x500+500+180")
self.bphigh=bphigh
self.pulse=pulse
    self.rbc=rbc
    self.wbc=wbc
self.cholesterol=cholesterol
self.hb=hb
    self.name=name
    self.age=age
self.master.resizable(0,0)
self.master.title("Report Generator")
self.master.configure(background="#332324")
self.title=Label(self.master, text="Generated Report",font=("Britannic Bold",22, "bold"), fg="white",
bg="#332324" ).place(x=130, y=15)
self.note=Label(self.master, text="->Enter name whose record you want to fetch<-", font=("Calibri",
10, "italic"),fg="white", bg="#332324").place(x=125, y=56)
self.lname=Label(self.master, text="Name", font=("Franklin Gothic Medium",12, "bold"), fg="white",
bg="#332324").place(x=105, y=100)
self.ename=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.ename.insert(END, self.name)

```

```

self.ename.place(x=250,y=100)
self.lage=Label(self.master, text="Age", font=("Franklin Gothic Medium",12, "bold"), fg="white",
bg="#332324").place(x=105, y=140)
self.eage=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.eage.insert(END, self.age )
self.eage.place(x=250,y=140)
self.lbphigh=Label(self.master, text="BP", font=("Franklin Gothic Medium",12, "bold"), fg="white",
bg="#332324").place(x=105, y=180)
self.ebphigh=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.ebphigh.insert(END, self.bphigh )
self.ebphigh.place(x=250,y=180)
self.lpulse=Label(self.master, text="Pulse Rate", font=("Franklin Gothic Medium",12, "bold"),
fg="white", bg="#332324").place(x=105, y=220)
self.epulse=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.epulse.insert(END, self.pulse )
self.epulse.place(x=250,y=220)
self.lrbcl=Label(self.master, text="RBC Count", font=("Franklin Gothic Medium",12, "bold"),
fg="white", bg="#332324").place(x=105, y=260)
self.erbc=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.erbc.insert(END, self.rbc )
self.erbc.place(x=250,y=260)
self.lcholesterol=Label(self.master, text="Cholesterol", font=("Franklin Gothic Medium",12, "bold"),
fg="white", bg="#332324").place(x=105, y=300)
self.echolesterol=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.echolesterol.insert(END, self.cholesterol )
self.echolesterol.place(x=250,y=300)
    self.lhb=Label(self.master, text="Haemoglobin", font=("Franklin Gothic Medium",12, "bold"),
fg="white", bg="#332324").place(x=105, y=340)
    self.ehb=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.ehb.insert(END, self.hb )
self.ehb.place(x=250,y=340)
self.lwbc=Label(self.master, text="WBC Count", font=("Franklin Gothic Medium",12, "bold"),
fg="white", bg="#332324").place(x=105, y=380)
self.ewbc=Text(self.master, width=20, height=1, bg="white", font=("Segoe UI Semibold", 12))
self.ewbc.insert(END, self.wbc )
self.ewbc.place(x=250,y=380)
self.back=Button(self.master, text="<- Back",font=("Berlin Sans FB",13), command=self.back,
fg="#332324", bg="white").place(x=420, y=425)
self.save=Button(self.master, text="Save to database",font=("Berlin Sans FB",13),width=15,
command=self.db, fg="#332324", bg="white")
self.save.place(x=222, y=425)

```

```

self.see=Button(self.master, text="Fetch record",font=("Berlin Sans FB",13),width=15,
command=self.show, fg="#332324", bg="white")
self.see.place(x=30, y=425)

def show(self):    #fetch record(displays record)
self.ip=self.ename.get("1.0",'end-1c')
y=x.execute('select * from people where name = "%s" ' % (self.ip))
count=0
for i in y:
    count+=1
self.eage.insert(END, i[1])
self.ebphigh.insert(END, i[2])
self.epulse.insert(END, i[3])
self.erbc.insert(END, i[4])
self.echolesterol.insert(END, i[5])
self.ehb.insert(END, i[6])
self.ewbc.insert(END, i[7])
if(count==0):
messagebox.showinfo("Alert", "No matching record found")
self.see.config(state=DISABLED)

def db(self):    #save to database
x.execute("create table if not exists people(name text, age text, bp text, pulse text, rbc text,
cholesterol text, haemoglobin text, wbc text)")
x.execute("insert into people values(?,?,?,?,?,?,?)", (self.name, self.age, self.bphigh, self.pulse,
self.rbc, self.cholesterol, self.hb, self.wbc))
x.commit()
self.save.config(text="Saved")
self.save.config(state=DISABLED)

def back(self):    #back button
self.master.destroy();

def main():    #main function
root=Tk()
welcomegui=welcome(root)
root.mainloop()

if __name__=="__main__":
    main();

```

Results

We finally got the end product as a 'fitness calculator' that includes all the mentioned modules. We learnt how to make a GUI using Tkinter in Python and also learnt to implement database connectivity using sqlite3.

This calculator offers the number of tools which helps you determine everything from your body mass index to your body fat composition, daily calorie need and more. Each of those tools will help you find an exercise program that fits your individual needs and goals. The following are the end results that we yielded from our project.

- Report generator will ultimately help the user to identify his/her fitness level. so, that they can take necessary steps to improve their physical condition.
- Body Mass Calculator will quantify the amount of tissue mass in an individual and the categorize the person as underweight, normal weight and overweight.
- Vo2 max will tell about the maximum rate of oxygen consumption.
- Calorie calculator will give the calorie count, which a person needs to consume each day.
- BFC will tell about the fat percentage in the body.
- BMR will tell the amount of energy needed while resting in a temperate environment when the digestive system is inactive.

References

- <https://www.calculator.net/fitness-and-health-calculator.html>
- <https://www.python-course.eu/index.php>
- www.blog.pythonlibrary.org
- <http://effbot.org/tkinterbook/>
- <http://www.dealingdata.net/2016/08/21/Python-MySQL-GUI/>
- <https://python-forum.io/index.php>
- <https://www.w3schools.com/python/>
- <https://www.geeksforgeeks.org/sql-using-python/>
- <https://stackoverflow.com/>
- www.tutorialspoint.com
- www.reddit.com
- www.google.co.in
- www.quora.com