

MINOR PROJECT REPORT

(Project Term SEP-NOV, 2020)

FITNESS CALCULATOR

Submitted by:

NAME OF STUDENT: Jaspreet Singh Kathuria

Registration Number: 11918806

NAME OF STUDENT: Puneet Kumar

Registration Number: 11908274

NAME OF STUDENT: Rashmi Kumari

Registration Number: 11908242

SECTION: K19GE

Course Code: INT213

Under the Guidance of

(Mr. Dipen Saini, Assistant Professor)

School of Computer Science and Engineering



**L OVELY
P ROFESSIONAL
U NIVERSITY**

DECLARATION

I hereby declare that the Mini project work entitled “FITNESS CALCULATOR” is an authentic record of our own work carried out in B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr.Dipen Saini during September To November 2020. All the information furnished in this mini project report is based on our own intensive work and is genuine.

NAME OF STUDENT: Jaspreet Singh Kathuria

Registration Number: 11918806

NAME OF STUDENT: Puneet Kumar

Registration Number: 11908274

NAME OF STUDENT: Rashmi Kumari

Registration Number: 11908242

(Signature of Student)

Date: 30-10-2020

CERTIFICATE

- This is to certify that the declaration statement made by the student is correct to the best of my knowledge and belief.
- He/she have completed this Mini Project under my guidance and supervision. The present work is the result of his/her original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Mini Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Mr. Dipen Saini

Assistant Professor,
School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date: 30-10-2020

ACKNOWLEDGEMENT

I take this opportunity to present my votes of thanks to all those guideposts who really acted as lightening pillars to enlighten my way throughout this project that has led to successful and satisfactory completion of this study.

I am grateful to Mr. Dipen Saini for providing me with an opportunity to undertake this project and providing me with all the facilities. I am highly thankful to sir for his active support, valuable time and advice, whole-hearted guidance, sincere cooperation and pains-taking involvement during the study and in completing the assignment of preparing the said project within the time stipulated.

Lastly, I am thankful to all those, particularly the various friends, who have been instrumental in creating proper, healthy and conducive environment and including new and fresh innovative ideas for me during the project, without their help, it would have been extremely difficult for me to prepare the project in a time bound framework.

INDEX

Declaration.....	i
Certificate.....	ii
Acknowledgement	iii
1. Introduction	1
2. Requirements of Project	2
3. Analysis of the Project.....	3-4
4. Coding of Project.....	5-11
5. Screen Shots of the project.....	11-12
6. Limitations of the Project (If any).....	13
7. Project Legacy	13
8. Gantt Chart.....	14
9. Bibliography.....	15

INTRODUCTION

Project allocated to our team is Fitness Calculator. It is for a Single User. It will generate report of single person at a time.

This Mini project of “Fitness Calculator” is purely made in python, with a good user-friendly interface which lets the user to enter the various values required to be input in order to calculate how fit the person is.

The main window consists of all the entry fields that the user is required to fill in order to generate a report. This main window also has a button to generate report when all the entry box has the required values.

PROBLEM STATEMENT:

This Program generates report of the particular person with following labels:

- BMI
- Blood Pressure
- Pulse Rate
- RBC, WBC, Platelets Count
- Haemoglobin level
- Uric Acid level
- Cholesterol level

and categorize it as high, low and normal.

These different values are colored according to how harmful they are on the individual's health. This makes it easier for the user to view the report, moreover “tk” module of tkinter library is used to show these values as a progress making it easier for the person using the app.

LIBRARIES USED:

This project uses tkinter library which is used to make the G.U.I. Some modules of tkinter used are buttons, entry box, labels and progress bars.

All here are used to represent the report graphically to the user.

REQUIREMENT OF PROJECT

Hardware and Software Requirements of Project

Hardware Requirement:

Here is the recommended hardware requirement for this software to run efficiently.

- 1) Intel core i3 or higher processor
- 2) 10 MB RAM
- 3) 15 MB free hard disc space
- 4) SVGA monitors / Laptops
- 5) Printer (Optional)

Software Requirement:

This software comes under application software. So, the necessary software for this is:

- 1) Windows operating system family.
- 2) Python Idle
- 3) MS Office

PROBLEM ANALYSIS

Program used a bunch of if statements used to decide the output according to the input given by user. This is the basic logic on which the program works.

The calculation of BMI and other values is done by using formulas and data collected from various websites is used to calculate the strength of a particular value in terms of High low or normal.

There are various functions and if statement which also control what colour to change for the label when a certain value is entered.

BASIS OF CALCULATION MODULES:

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height (m}^2\text{)}}$$

BMI	Weight class
below 18.5	underweight
18.5 - 24.9	normal
25.0 - 29.9	overweight
30.0 and up	Very overweight

Blood Pressure Categories

BLOOD PRESSURE CATEGORY	SYSTOLIC mm Hg (upper number)		DIASTOLIC mm Hg (lower number)
NORMAL	LESS THAN 120	and	LESS THAN 80
ELEVATED	120 – 129	and	LESS THAN 80
HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 1	130 – 139	or	80 – 89
HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 2	140 OR HIGHER	or	90 OR HIGHER
HYPERTENSIVE CRISIS (consult your doctor immediately)	HIGHER THAN 180	and/or	HIGHER THAN 120

Age	Normal Pulse Rate
Neonate (<28 days)	100 - 205
Infant (1 month – 1 year)	100 - 190
Toddler (1 – 2 years)	98 - 140
Pre-school (3 – 5 years)	80 - 120
6 – 11 years	75 - 118
12 years – adult	60 – 100
Athlete	40 - 60

Table 5. Normal ranges of full blood count. (SI unit)

Parameter	Normal Range (Male)	Normal Range (Female)
Red Blood Cell (RBC)	$4.50 - 6.50 \times 10^{12} /L$	$3.80 - 5.80 \times 10^{12} /L$
White Blood Cell (WBC)	$3.60 - 11.0 \times 10^9 /L$	$3.60 - 11.0 \times 10^9 /L$
Platelet	$150 - 450 \times 10^9 /L$	$150 - 450 \times 10^9 /L$

mg/dL	μmol/L	mmol/L	Diagnosis
5 or less	300 or less	.30 or less	Safe
5 – 6	300 – 350	.30 – .35	Good
6 – 7	350 – 400	.35 – .40	Warning
Over 7	Over 400	Over .40	Danger
Vital Uric Acid Levels Conversion Chart			

	Low Normal Range	High Normal Range	Acceptable for Donating
Males	13.5g/dL	17.5g/dL	13.0g/dL
Females	12.0g/dL	15.5g/dL	12.5g/dL

THE CODE

```
from tkinter import *

import tkinter

import tkinter.messagebox

def main():

    top1=Tk()

    top1.geometry('400x300')

    name=str(E1.get())

    L00 = Label(top1, text=name + " Your Medical Report is:",fg="brown",font="Times").grid(row=0,column=0)

    """ _____BMI_____ """

    l12=Label(top1,text="BMI:",fg="red")

    l12.grid(row=1,column=0)

    E12=Entry(top1,bd=5)

    E12.grid(row=1,column=1)

    E122=Entry(top1,bd=5)

    E122.grid(row=1,column=2)

    weight1=int(E3.get())

    height1=float(E4.get())

    age1=int(E2.get())

    bmi1=weight1/(height1*height1)

    E12.insert(0,bmi1)

    E12['bg']='orange'

    if (bmi1<18.5):

        E122.insert(0,"Underweight")

        E122['bg']='red'

    elif (bmi1>=18.5 and bmi1<=24.9):

        E122.insert(0,"Normal")

        E122['bg']='yellow'

    elif (bmi1>=25.0 and bmi1<=29.9):

        E122.insert(0,"Overweight")

        E122['bg']='grey'

    else:

        E122.insert(0,"Very Overweight")
```

```

E122['bg']='green'

"""          BP          """

L11=Label(top1,text="BP(HIGH/NORMAL/LOW):",fg="red")

L11.grid(row=2,column=0)

E11=Entry(top1,bd=5)

E11.grid(row=2,column=1)

bplow1=int(E5.get())

if (bplow1<=90):

    E11.insert(0,"Low")

    E11['bg']='red'

elif (bplow1>90 and bplow1<120):

    E11.insert(0,"Normal")

    E11['bg']='yellow'

else:

    E11.insert(0,"High")

    E11['bg']='green'

"""          Pulserate          """

L22=Label(top1,text="Pulse Rate(HIGH/NORMAL/LOW):",fg="red")

L22.grid(row=3,column=0)

E22=Entry(top1,bd=5)

E22.grid(row=3,column=1)

pulserate1=int(E6.get())

if(pulserate1<60):

    E22.insert(0,"low")

    E22['bg']='red'

elif (pulserate1>60 and pulserate1<100):

    E22.insert(0,"Normal")

    E22['bg']='yellow'

else:

    E22.insert(0,"High")

    E22['bg']='green'

"""          RBC count          """

L33=Label(top1,text="RBC Count(HIGH/NORMAL/LOW):",fg="red")

L33.grid(row=5,column=0)

```

```

E33=Entry(top1,bd=5)

E33.grid(row=5,column=1)

rbccount1=int(E7.get())

if(rbccount1<475000):

    E33.insert(0,"Low")

    E33['bg']='red'

elif(rbccount1>475000 and rbccount1<610000):

    E33.insert(0,"Normal")

    E33['bg']='yellow'

else:

    E33.insert(0,"High")

    E33['bg']='green'

""""_____WBC COUNT_____""""

L44=Label(top1,text="WBC Count(HIGH/NORMAL/LOW):",fg="red")

L44.grid(row=6,column=0)

E44=Entry(top1,bd=5)

E44.grid(row=6,column=1)

wbccount1=int(E8.get())

if(wbccount1<4000):

    E44.insert(0,"Low")

    E44['bg']='red'

elif(wbccount1>4000 and wbccount1<10000):

    E44.insert(0,"Normal")

    E44['bg']='yellow'

else:

    E44.insert(0,"High")

    E44['bg']='green'

""""_____PLALATES_____""""

L55=Label(top1,text="Platelets(HIGH/NORMAL/LOW):",fg="red")

L55.grid(row=7,column=0)

E55=Entry(top1,bd=5)

E55.grid(row=7,column=1)

platelets1=int(E9.get())

if(platelets1<150000):

```

```

E55.insert(0,"Low")

E55['bg']='red'

elif (platelets1>150000 and platelets1<450000):

E55.insert(0,"Normal")

E55['bg']='yellow'

else:

E55.insert(0,"High")

E55['bg']='green'

""" _____HEMOGLOBIN_____ """

L66=Label(top1,text="HB(HIGH/NORMAL/LOW):",fg="red")

L66.grid(row=8,column=0)

E66=Entry(top1,bd=5)

E66.grid(row=8,column=1)

hb1=int(E10.get())

if(hb1<12):

E66.insert(0,"Low")

E66['bg']='red'

elif(hb1>12 and hb1<16):

E66.insert(0,"Normal")

E66['bg']='yellow'

else:

E66.insert(0,"High")

E66['bg']='green'

""" _____URIC ACID_____ """

L77=Label(top1,text="Uric Acid(HIGH/NORMAL/LOW):",fg="red")

L77.grid(row=9,column=0)

E77=Entry(top1,bd=5)

E77.grid(row=9,column=1)

uricacid1=int(E50.get())

if(uricacid1<4):

E77.insert(0,"Low")

E77['bg']='red'

elif(uricacid1>4 and uricacid1<7):

E77.insert(0,"Normal")

```

```

E77['bg']='yellow'

else:

E77.insert(0,"High")

E77['bg']='green'

""" _____CHOLESTROL_____ """

L88=Label(top1,text="Cholestrol(HIGH/NORMAL/LOW):",fg="red")

L88.grid(row=10,column=0)

E88=Entry(top1,bd=5)

E88.grid(row=10,column=1)

cholesterol1=int(E51.get())

if(cholesterol1<40):

E88['bg']='red'

E88.insert(0,"low")

elif(cholesterol1>40 and cholesterol1<50):

E88.insert(0,"Normal")

E88['bg']='yellow'

else:

E88.insert(0,"High")

E88['bg']='green'

top1.mainloop()

top = Tk()

L0 = Label(top, text="Fitness Calculator",fg="purple",font="Times").grid(row=0,column=1)

L1=Label(top,text="Name",fg="red",font="Times").grid(row=1,column=0)

E1= Entry(top,bd=5)

E1.grid(row=1,column=1)

L2=Label(top,text="age ",fg="red",font="Times").grid(row=2,column=0)

E2= Entry(top,bd=5)

E2.grid(row=2,column=1)

l3=Label(top,text="weight (in Kg)",fg="red",font="Times").grid(row=3,column=0)

E3= Entry(top,bd=5)

E3.grid(row=3,column=1)

l4=Label(top,text="Height (in M)",fg="red",font="Times").grid(row=4,column=0)

E4=Entry(top,bd=5)

E4.grid(row=4,column=1)

```

```

l5=Label(top,text="Bp (0-120)",fg="red",font="Times").grid(row=5,column=0)

E5=Entry(top,bd=5)

E5.grid(row=5,column=1)

L6=Label(top,text="Pulserate (0-100)",fg="red",font="Times").grid(row=6,column=0)

E6=Entry(top,bd=5)

E6.grid(row=6,column=1)

L7=Label(top,text="RBCcount ( 310000-610000)",fg="red",font="Times").grid(row=7,column=0)

E7 = Entry(top,bd=5)

E7.grid(row=7,column=1)

L8=Label(top,text="WBCcount (2000-10000)",fg="red",font="Times").grid(row=8,column=0)

E8= Entry(top,bd=5)

E8.grid(row=8,column=1)

L9=Label(top,text="Platelets (150000-615000)",fg="red",font="Times").grid(row=10,column=0)

E9 = Entry(top,bd=5)

E9.grid(row=10,column=1)

L10=Label(top,text="HEMOGLOBIN(0-16)",fg="red",font="Times").grid(row=11,column=0)

E10 = Entry(top,bd=5)

E10.grid(row=11,column=1)

L50=Label(top,text="URIC ACID (0-7)",fg="red",font="Times").grid(row=12,column=0)

E50 = Entry(top,bd=5)

E50.grid(row=12,column=1)

L51=Label(top,text="CHOLESTROL(40-55)",fg="red",font="Times").grid(row=13,column=0)

E51 = Entry(top,bd=5)

E51.grid(row=13,column=1)

def clear_textbox():

    E1.delete(0,END)

    E2.delete(0, END)

    E3.delete(0, END)

    E4.delete(0, END)

    E5.delete(0, END)

    E6.delete(0, END)

    E7.delete(0, END)

    E8.delete(0, END)

    E9.delete(0, END)

```

```

E10.delete(0, END)

E50.delete(0, END)

E51.delete(0, END)

def close_window ():

    top.destroy()

D=Label(top,text="Developed By:-",fg='green')

D.grid(row=19,column=0)

D1=Label(top,text="Rashmi Kumari",fg='brown')

D1.grid(row=19,column=1)

D2=Label(top,text="Jaspreet Singh",fg='red')

D2.grid(row=19,column=2)

D2=Label(top,text="Puneet",fg='black')

D2.grid(row=19,column=3)

B14=Button(top, text='RESET ALL ENTRIES',command=clear_textbox,bg="cyan")

B14.grid(row=14,column=0)

B15=Button(top,text = "EXIT",command=close_window,bg="yellow")

B15.grid(row=14,column=2)

B1=Button(top, text='Generate Report',command=main,bg="orange")

B1.grid(row=13,column=2)

top.mainloop()

```

SCREEN SHOTS

FUTURE SCOPE AND LIMITATION

Creating worldwide health platform to store and generate report of a person's health. If someone's health goes bad on this platforms, different health organizations aka Hospitals can tap that person and help him to get to know the symptoms.

As the Future, the young generation of the world we have to take care of our health. Knowing the disease faster makes us having more chances of curing it more efficiently.

Though the system has been designed according to the requirements of the users it has its own limitation. Thus, the **Limitation** of system is: -

- 1.) No facility to generate report for multiple users.
- 2.) Data Used for Calculation is collected from various website. So, possibility of incorrectness of values.
- 3.) Data is not stored in the memory.

PROJECT LEGACY

Being Mental, Emotional, Physically Fit and aware is extremely important in this age. Pollution, sickness is one of the main causes of death today.

Reaching and maintaining a healthy body is important for overall health and can help you prevent and control many diseases and conditions. If your levels are LOW or HIGH, you are at high risk of developing serious health problems, including heart disease, high blood pressure, type 2 diabetes, gallstones, breathing problems, and certain cancers. That is why maintaining a healthy body is so important: It helps you lower your risk for developing these problems, helps you feel good about yourself, and gives you more energy to enjoy life.

Gantt chart:

We planned our project according to the Gantt chart as follows:

Sr. No.	Task	Start	Finish	Duration (in Days)
1	Information gathering	20-09-2020	27-09-2020	7
2	Database design	28-09-2020	04-10-2020	6
3	Form design	05-10-2020	10-10-2020	5
4	Coding	11-10-2020	20-10-2020	9
5	Testing	21-10-2020	23-10-2020	2
6	Implementation	23-10-2020	29-11-2020	6

BIBLIOGRAPHY

1. *<https://www.tutorialspoint.com/python>*
2. *<https://docs.python.org/3/tutorial>*
3. *www.programiz.com/python-programming*
4. *https://en.wikibooks.org/wiki/A_Beginner's_Python_Tutorial*
5. Quora.com