

PYTHON MINI PROJECT REPORT

Project No: 24
To Design a GUI for Fitness Calculator
using python.



Submitted By

1. UTKARSH SINGH (RK21RGA74) (11603042)
2. RUSHIK PATEL (RK21RGB49) (12112228)
3. AVNEESH NARWAL (RK21RGA24) (12107455)

Table Of Contents

Sr.no	Topics	Page. No
1	Introduction	3
2	Libraries Used	4
3	Hardware	4
4	Logic Used	4
5	Source Code	8
6	Snap Shots	16
7	Future Scope	19
8	Conclution	19

Introduction

Project allocated to our team is Fitness Calculator. It is a Single User 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. Program which generates report of the particular person with following labels:

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

and categorize it as high, low and medium.

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. We have also used file management to provide some information about the term in report.

HARDWARE

As a part of hardware, the user should have a laptop or desktop computer which is capable of handling python idle. It should also have minimum of 2GB ram for the easy compilation and smooth running of the program.

LOGIC USED

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 medium.

BASIS OF CALCULATION MODULES

For BMI: -

Body mass index is a value derived from the mass and height of a person.

The BMI is defined as the body mass divided by the square of the body height and is expressed in units of kg/m².

If your BMI is 18.5 to 24.9, it falls within the normal or Healthy Weight range.

If your BMI is 25.0 to 29.9, it falls within the overweight range.

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

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

For Low blood pressure: -

Low blood pressure is a reading of less than 90/60mmHg. It does not always cause symptoms, but you may need treatment if it does.

For High blood pressure: -

Some health care professionals diagnose patients with high blood pressure if their blood pressure is consistently 140/90 mm Hg or higher.

For Pulse Rate: -

The normal pulse for healthy adult's ranges from 60 to 100 beats per minute. The pulse rate may fluctuate and increase with exercise, illness, injury, and emotions. Females ages 12 and older, in general, tend to have faster heart rates than do males.

For RBC count: -

A normal Red Blood Cell count would be around:

- men = 4.0 to $5.9 \times 10^{12}/L$,
- women = 3.8 to $5.2 \times 10^{12}/L$.

For WBC count: -

The normal number of White Blood Cell in the blood is 4,500 to 11,000 WBCs per microliter (4.5 to $11.0 \times 10^9/L$).

For Platelets: -

They are small, colorless cell fragments in our blood that form clots and stop or prevent bleeding. A normal platelet count in adults ranges from 150,000 to 450,000 platelets per microliter of blood.

For Hemoglobin: -

It is a protein in your red blood cells that carries oxygen to your body's organs and tissues, it transports carbon dioxide from your organs and tissues back to your lungs.

The healthy range for hemoglobin is:

- Men = 13.2 to 16.6 grams/ deciliter,
- Women = 11.6 to 15 grams/ deciliter.

For Uric Acid: -

It is a waste product found in blood. It's created when the body breaks down chemicals called purines. Most uric acid dissolves in the blood, passes through the kidneys and leaves the body in urine. Normal values range between 3.5 to 7.2 milligrams per deciliter (mg/dL).

For Cholesterol: -

Average cholesterol in adults:

- Normal = Less than 200 mg/dL.
- Borderline high = 200 to 239 mg/dL.
- High = At or above 240 mg/dL.

According to the above mentioned criteria we generate fitness report whether the particular category falls in healthy range or not. And according to that it gives you that whether your mentioned details high, normal, or less.

You also get this information in the calculator from the menu bar, go to help and click on than information button. It will open one text file containing all the above information. Also in help there

SOURCE CODE

```
from tkinter import *
import os
import tkinter as tk

def btn_clicked():
    class FitnessCalculator:
        def __init__(self):

            calculator = tk.Toplevel(win)
            calculator.title('Fitness Calculator')
            calculator.resizable(False, False)

            mainmenu = Menu(calculator, background='#76a9c1', fg='white')

            def About():
                print("Let me tell you About us \n")
                os.startfile("ABOUT.txt")

            def REC():
                print("Showing User Lists \n")
                os.startfile("RECORDS.txt")

            def Help():
                print("Happy to Help \n")
                os.startfile("HELP.txt")
                #f = open("C:\\Users\\hp\\Desktop\\study\\PYTHON\\code\\New
folder\\Final\\Help.txt",'r')
                # print(f.read())

            def myfunc():
                print("HELLO")
            m1 = Menu(mainmenu, tearoff=0, background='grey', fg='white')
            m1.add_command(label="New", command=myfunc)
            m1.add_command(label="Save", command=myfunc)
            m1.add_command(label="Open", command=REC)
            m1.add_separator()
            m1.add_command(label="Exit", command=calculator.destroy)
            calculator.config(menu=mainmenu)
            edit = Menu(mainmenu, tearoff=0, background='grey', fg="white")
            mainmenu.add_cascade(label='Edit', menu=edit)
            edit.add_command(label='Cut', command=None)
            edit.add_command(label='Copy', command=None)
            edit.add_command(label='Paste', command=None)
            edit.add_command(label='Select All', command=None)
            edit.add_separator()
            edit.add_command(label='Find...', command=None)
            edit.add_command(label='Find again', command=None)
            mainmenu.add_cascade(label="File", menu=m1)
            m2 = Menu(mainmenu, tearoff=0, background='grey', fg="white")
            m2.add_command(label="About us", command=About)
            m2.add_separator()
```



```

m2.add_command(label="Information", command=Help)
calculator.config(menu=mainmenu)
mainmenu.add_cascade(label="Help", menu=m2)

# Name and Age Entry Fields
name = StringVar()
age = IntVar()
Label(calculator, text='Name : ', width=10).grid(row=0, column=1)
Entry(calculator, width=25, relief='ridge', bd=2,
      textvariable=name).grid(row=0, column=2)
Label(calculator, text='Age : ', width=10).grid(row=1, column=1)
Entry(calculator, width=25, relief='ridge', bd=2,
      textvariable=age).grid(row=1, column=2)

# The Gender Radio Button
v = IntVar()
Label(calculator, text='Gender : ').grid(row=2, column=1)
Radiobutton(calculator, text='Male', variable=v,
            value=1).grid(row=2, column=2)
Radiobutton(calculator, text='Female', variable=v,
            value=2).grid(row=2, column=4)

# Generating the fields for Entry Fields
fields = ('Weight (Kg)', 'Height (Mts)', 'BP Low (Systolic) mm/Hg', 'BP High
(Diastolic) mm/Hg', 'Pulse Rate',
          'RBC Count (trillion Cells/L)', 'WBC Count (billions cells/L)',
          'Platelets', 'HB', 'Uric Acid (mg/dL)', 'Cholestrol (mg/dL)')
r = 3
entries = []
for field in fields:
    Label(calculator, text=field+ ' : ', width=30, bg='grey',
          fg='white').grid(row=r, column=1, sticky='NWSWSE')
    en = Entry(calculator, width=30, relief='ridge', bd=2)
    en.grid(row=r, column=2, sticky='NWSWSE')
    entries.append(en)
    r += 1

# Report Label
Label(calculator, text='Report Of', width=30, bg='black',
      fg='white').grid(row=14, column=1, sticky='NWSWSE')
Label(calculator, text='BMI (High/Medium/Low)', width=30, fg='black',
      bg='gainsboro', textvariable=name).grid(row=14, column=2, sticky='NWSWSE')

# Results TextVariables
bmi_calculated = StringVar()
bp_calc = StringVar()
pulse_calculated = StringVar()
cholesterol_calculated = StringVar()
wbc_final = StringVar()
rbc_final = StringVar()
platelets_final = StringVar()
uric_acid = StringVar()
haemoglobin_calc = StringVar()

# Final_Labels

```

```

Label(calculator, text='BMI (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=15, column=1, sticky='NWSWSE')
Label(calculator, text='BP (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=16, column=1, sticky='NWSWSE')
Label(calculator, text='Pulse Rate (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=17, column=1, sticky='NWSWSE')
Label(calculator, text='RBC Count (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=18, column=1, sticky='NWSWSE')
Label(calculator, text='WBC Count (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=19, column=1, sticky='NWSWSE')
Label(calculator, text='Platelets (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=20, column=1, sticky='NWSWSE')
Label(calculator, text='HB (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=21, column=1, sticky='NWSWSE')
Label(calculator, text='Uric Acid (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=22, column=1, sticky='NWSWSE')
Label(calculator, text='Cholestrol (High/Medium/Low)', width=30,
      bg='grey', fg='white').grid(row=23, column=1, sticky='NWSWSE')

Label(calculator, text='BMI (High/Medium/Low)', width=30, fg='black',
bg='gainsboro',
      textvariable=bmi_calculated).grid(row=15, column=2, sticky='NWSWSE')
Label(calculator, text='BP (High/Normal/Low)', width=30, fg='black',
      bg='gainsboro', textvariable=bp_calc).grid(row=16, column=2,
sticky='NWSWSE')
Label(calculator, text='Pulse Rate (High/Medium/Low)', width=30,
bg='gainsboro',
      fg='black', textvariable=pulse_calculated).grid(row=17, column=2,
sticky='NWSWSE')
Label(calculator, text='RBC Count (High/Medium/Low)', width=30,
bg='gainsboro',
      fg='black', textvariable=rbc_final).grid(row=18, column=2,
sticky='NWSWSE')
Label(calculator, text='WBC Count (High/Medium/Low)', width=30,
bg='gainsboro',
      fg='black', textvariable=wbw_final).grid(row=19, column=2,
sticky='NWSWSE')
Label(calculator, text='Platelets (High/Medium/Low)', width=30,
bg='gainsboro',
      fg='black', textvariable=platelets_final).grid(row=20, column=2,
sticky='NWSWSE')
Label(calculator, text='HB (High/Medium/Low)', width=30, fg='black',
bg='gainsboro',
      textvariable=haemoglobin_calc).grid(row=21, column=2, sticky='NWSWSE')
Label(calculator, text='Uric Acid (High/Medium/Low)', width=30,
bg='gainsboro',
      fg='black', textvariable=uric_acid).grid(row=22, column=2,
sticky='NWSWSE')
Label(calculator, text='Cholestrol (High/Medium/Low)', width=30,
bg='gainsboro',
      fg='black', textvariable=cholestrol_calculated).grid(row=23, column=2,
sticky='NWSWSE')

```

```
# making the bottom Label
```

```

Label(calculator, bg='black', fg='white',
      width=30).grid(row=25, column=1, padx=5)

def get_entries():
    print(name.get())
    print(age.get())
    for entry in entries:
        print(entry.get())

# BMI calculator

    def calculate_bmi():
        weight = float(entries[0].get())
        height = float(entries[1].get())
        bmi = (weight/height)/height
        if bmi < 15:
            value = 'Low'
        elif bmi > 15 and bmi < 25:
            value = 'Medium'
        else:
            value = 'High'
        bmi_calculated.set(value)

# BP monitor

def blood_pressure():
    bpl = float(entries[2].get())
    bph = float(entries[3].get())
    if bpl < 120 and bph < 80:
        value = 'Normal'
    else:
        value = 'High'
    bp_calc.set(value)

# Pulse Monitor

def pulse_rate():
    pulse = int(entries[4].get())
    if pulse < 72:
        value = 'Low'
    elif pulse > 72 and pulse < 90:
        value = 'Medium'
    else:
        value = 'High'
    pulse_calculated.set(value)

# Blood Monitor

def blood_monitor():
    rbc = float(entries[5].get())
    wbc = float(entries[6].get())
    platelets = float(entries[7].get())
    haemoglobin = float(entries[8].get())

```

```

# calculating Haemoglobin Levels

    if haemoglobin < 13:
        h_value = 'Low'
    elif haemoglobin > 13 and haemoglobin < 16:
        h_value = 'Medium'
    else:
        h_value = 'High'
# calculating rbc levels

    if rbc < 4.32:
        r_value = 'Low'
    if rbc > 4.32 and rbc < 5.72:
        r_value = 'Medium'
    else:
        r_value = 'High'

# calculating wbc values

    if wbc < 3:
        w_value = 'Low'
    if wbc > 3 and wbc < 10:
        w_value = 'Medium'
    else:
        w_value = 'High'
# calculating platelets levels

    if platelets > 150 and platelets < 450:
        p_value = 'Medium'
    elif platelets < 150:
        p_value = 'Low'
    else:
        p_value = 'High'

# setting Label Values
    haemoglobin_calc.set(h_value)
    rbc_final.set(r_value)
    wbc_final.set(w_value)
    platelets_final.set(p_value)

# urine Monitor

    def urine_monitor():
        urine = float(entries[9].get())
        if urine < 4:
            value = 'Low'
        elif urine > 4 and urine < 8.5:
            value = 'Medium'
        else:
            value = 'High'
        uric_acid.set(value)

# get Cholestrol

```

```

def get_cholesterol():
    cholesterol = float(entries[10].get())
    if cholesterol < 200:
        value = 'Low (Good)'
    elif cholesterol > 200 and cholesterol < 239:
        value = 'Medium'
    else:
        value = 'High'
    cholesterol_calculated.set(value)

def show_results():
    name.set(name.get())
    get_entries()
    calculate_bmi()
    blood_pressure()
    pulse_rate()
    blood_monitor()
    urine_monitor()
    get_cholesterol()

def Log():
    window2 = tk.Toplevel(calculator)
    window2.title("Log-IN / Sign-UP")

    window2.geometry("1000x600")
    window2.configure(bg="#aaaaaa")
    canvas = Canvas(
        window2,
        bg="#aaaaaa",
        height=600,
        width=1000,
        bd=0,
        highlightthickness=0,
        relief="ridge")
    canvas.place(x=0, y=0)

    background_img = PhotoImage(
        file=f"bg.png")
    background = canvas.create_image(
        500.0, 300.0,
        image=background_img)

    entry0_img = PhotoImage(
        file=f"img_textBox0.png")
    entry0_bg = canvas.create_image(
        616.0, 314.5,
        image=entry0_img)
    E0 = StringVar()
    E1 = StringVar()
    E2 = StringVar()
    entry0 = Entry(window2, textvariable=E0,
        bd=0,
        bg="#76a9c1",
        highlightthickness=0)

```

```

entry0.place(
    x=448, y=284,
    width=336,
    height=59)

entry1_img = PhotoImage(
    file=f"img_textBox1.png")
entry1_bg = canvas.create_image(
    616.0, 175.5,
    image=entry1_img)

entry1 = Entry(window2, textvariable=E1,
    bd=0,
    bg="#76a9c1",
    highlightthickness=0)

entry1.place(
    x=448, y=145,
    width=336,
    height=59)

entry2_img = PhotoImage(
    file=f"img_textBox2.png")
entry2_bg = canvas.create_image(
    616.0, 452.5,
    image=entry2_img)

entry2 = Entry(window2, textvariable=E2,
    bd=0,
    bg="#76a9c1",
    highlightthickness=0)

entry2.place(
    x=448, y=422,
    width=336,
    height=59)

def btn_clicked():
    print(f"{E1.get(), E0.get(), E2.get()}")
    with open("RECORDS.txt", "a") as f:
        f.write(f"{E1.get(), E0.get(), E2.get()} \n")
    window2.destroy()

img0 = PhotoImage(
    file=f"0.png")
b0 = Button(window2,
    image=img0,
    borderwidth=0,
    highlightthickness=0,
    cursor="arrow",
    command=btn_clicked,
    relief="flat")
b0.place(

```

```

        x=839, y=521,
        width=131,
        height=49)

    window2.resizable(True, True)
    window2.mainloop()

    # Button to Show the results of the report
    Button(calculator, text="Show Report", relief='ridge', bg='grey',
           fg='white', bd=2, command=show_results).grid(row=14, column=4, ipadx=4)
    Button(calculator, text="Log-IN", relief='ridge', bg='grey',
           fg='white', bd=2, command=Log).grid(row=25, column=4, ipadx=4)
    # Initiating the GUI mainloop Instance (event loop)
    calculator.mainloop()

# calling the default constructor to create the GUI
FitnessCalculator()

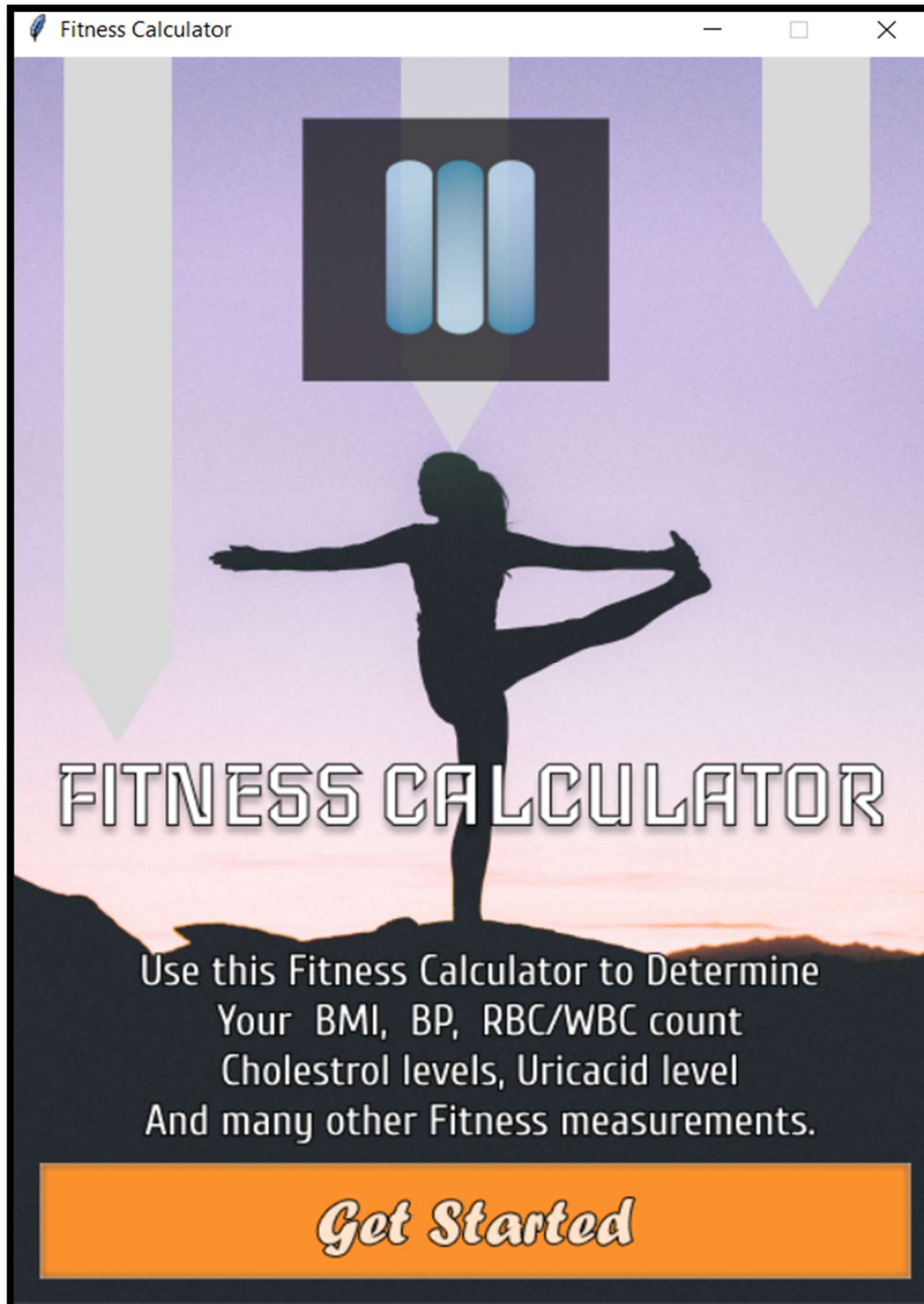
win = Tk()
win.title("Fitness Calculator")
win.geometry("496x700")
win.configure(bg="#ffffff")
canvas = Canvas(
    win,
    bg="#ffffff",
    height=700,
    width=496,
    bd=0,
    highlightthickness=0,
    relief="ridge")
canvas.place(x=0, y=0)
background_img = PhotoImage(
    file=f"background.png")
background = canvas.create_image(
    247.5, 298.0,
    image=background_img)
img0 = PhotoImage(
    file=f"img0.png")
b0 = Button(win,
            image=img0,
            borderwidth=0,
            highlightthickness=0,
            cursor="arrow",
            command=btn_clicked,
            relief="flat")
b0.place(
    x=17, y=621,
    width=468,
    height=65)

win.resizable(False, False)
win.mainloop()

```

USER INTERFACE OF CALCULATOR

Landing window: -




Calculator Interface before generating report: -

The screenshot shows a window titled "Fitness Calculator" with a menu bar containing "Edit", "File", and "Help". The interface is divided into several sections:

- Input Fields:**
 - Name :** A text input field.
 - Age :** A text input field containing the value "0".
 - Gender :** Two radio buttons labeled "Male" and "Female".
 - Weight (Kg) :** A text input field.
 - Height (Mts) :** A text input field.
 - BP Low (Systolic) mm/Hg :** A text input field.
 - BP High (Diastolic) mm/Hg :** A text input field.
 - Pulse Rate :** A text input field.
 - RBC Count (trillion Cells/L) :** A text input field.
 - WBC Count (billions cells/L) :** A text input field.
 - Platelets :** A text input field.
 - HB :** A text input field.
 - Uric Acid (mg/dL) :** A text input field.
 - Cholestrol (mg/dL) :** A text input field.
- Report Section:**
 - A header "Report Of" is positioned above a list of health metrics: BMI (High/Medium/Low), BP (High/Medium/Low), Pulse Rate (High/Medium/Low), RBC Count (High/Medium/Low), WBC Count (High/Medium/Low), Platelets (High/Medium/Low), HB (High/Medium/Low), Uric Acid (High/Medium/Low), and Cholestrol (High/Medium/Low).
 - To the right of this list is a large, empty rectangular area for the report output.
- Buttons:**
 - A "Show Report" button is located to the right of the report output area.
 - A "Log-IN" button is located at the bottom right of the window.

Calculator Interface after generating report: -


Fitness Calculator
—
□
×

Edit
File
Help

Name :
rushik

Age :
18

Gender :
☒ Male
☐ Female

Weight (Kg) :
75

Height (Mts) :
1.85

BP Low (Systolic) mm/Hg :
95

BP High (Diastolic) mm/Hg :
120

Pulse Rate :
77

RBC Count (trillion Cells/L) :
5

WBC Count (billions cells/L) :
4

Platelets :
300000

HB :
14

Uric Acid (mg/dL) :
5

Cholestrol (mg/dL) :
250

Report Of

rushik

Show Report

BMI (High/Medium/Low)

Medium

BP (High/Medium/Low)

High

Pulse Rate (High/Medium/Low)

Medium

RBC Count (High/Medium/Low)

Medium

WBC Count (High/Medium/Low)

Medium

Platelets (High/Medium/Low)

High

HB (High/Medium/Low)

Medium

Uric Acid (High/Medium/Low)

Medium

Cholestrol (High/Medium/Low)

High

Log-IN

FUTURE SCOPE

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.

CONCLUSION

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.