Python Project on –

Health Tracker and Recommendation System

END TERM REPORT

by

Anurag Anand(11901877)
Sushant Kumar(11902398)
Pabitra Kr. Panda(11903542)
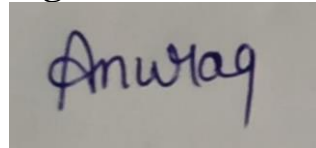
K19PV-A

Roll numbers - 09,15,75

Department of Intelligent Systems,
School of Computer Science Engineering,
Lovely Professional University, Jalandhar
November, 2020

Student Declaration

This is to declare that this report has been written by Anurag anand, Sushant Kumar and Pabitra kr. panda. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

Signature:

Name: Anurag Anand

Roll Number: 11901877

Place: Phagwara
Date: 31st Oct 2020

TABLE OF CONTENTS

BONAFIDE CERTIFICATE

Certified that this project report "Health tracker and Recommender" is the bonafide work of "Anurag Anand, Sushant Kumar and Pabitra kumar Panda" who carried out the project work under my supervision.

Signature:

Name: Dr. Dhanpratap Singh

Designation:

UMS ID: 25706

Department: School of computer science and engineering

BACKGROUND AND OBJECTIVE

Health recommendation system plays an important role in an individual life, it doesn't matter whether he/she is rich or poor. A good and cheap health recommendation can save a lots of rush, money and time. This application can even save lots of life by providing the proper guidance.

Sophisticated recommendation systems are used more and more in the health sector to assist consumers in healthy decision making. In this study we investigate consumers' evaluation of hypothetical health recommendation systems that provide personalized nutrition advice. We examine consumers' intention to use such a health recommendation system as a function of options related to the underlying system (e.g. the type of company that generates the advice) as well as intermediaries (e.g. general practitioner) that might assist in using the system. We further explore if the effect of both the system and intermediaries on intention to use a health recommendation system are mediated by consumers' perceived effort, privacy risk, usefulness and enjoyment.

Most of the people even know about the some common solution but they don't want to follow that for example they know that drinking is beneficial for health but they don't how that what's the real power, healing and prevention capacity of water is. This can be sorted if we can assure them that they are taking guidance from some the experts with all the proper reason.

This project helps the users in curing its disease by giving the list of fruits and herbs that the user should consume in order to get rid of its disease. The main purpose of this project is to help the user to easily search for herbs and fruits that will be good for the health of the user depending on any health issue or disease that he/she is suffering from. This system helps the

user to reduce its searching time to a great extent by allowing the user to enter its health problem and search accordingly. The admin can add fruits and herbs to the system and its information. This system also allows the user to view the selected fruit or the herb's description which describes how the fruit or the herb will help to improve the user's health. This system also allows the user to place order which will add the items to the user's cart and make payment for the same. The system also includes a module in which the user can search for the hospitals depending on the name of the disease that user enter. Thus, this system helps to get food products best suited for user health to a great extent. System can recommend some product to the user.

Some advantages of this system–

- This system helps the users to improve their health to a great extent depending on the health issue or the disease that the user is suffering from.
- This system can help to follow the proper diet for different medical conditions.
- This system also helps to identify your blood pressure, Sugar level and body mass index.
- This system reduces the search time to a great extent.
- System also recommends products to the user.

Goals and Objective –

We find evidence that the options related to the underlying system as well as the intermediaries involved influence consumers' intention to use such a health recommendation system and that these effects are mediated by perceptions of effort, privacy risk, usefulness and enjoyment. Also, we find that consumers value usefulness of a system more and enjoyment less when a general practitioner advices them to use a health recommendation system than if they use it out of their own curiosity.

We developed and tested a model of consumers' intention to use a health recommendation system. We found that intermediaries play an important role in how consumers evaluate such a system over and above options of the underlying system that is used to generate the recommendation. Also, health-related information services seem to rely on endorsement by the medical sector. This has considerable implications for the distribution as well as the communication channels of health recommendation systems which may be quite difficult to put into practice outside traditional health service channels.

PROJECT DESCRIPTIONS

The most first page of GUI(Graphical user interface) of our project is some more looking like this image pasted below.
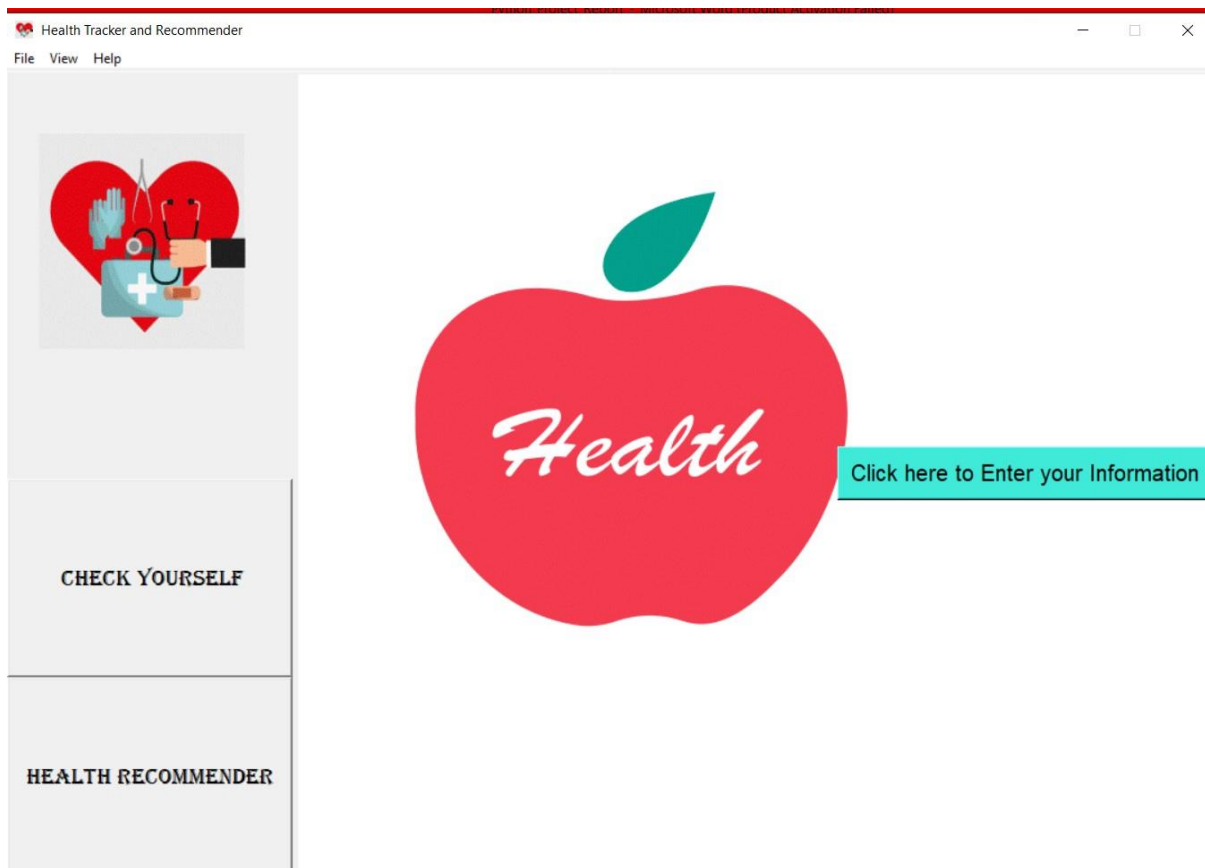
In this page of interface we have created 2 frame where the left side of the frame is having slider feature for users convenient. The right frame is having an option of **"Click here to enter your Information"** which allows user to enter their personal information just like a signup/login page, that helps to generate final report. And the right side of this interface is holding up 2 options **"Check yourself "** and **" Health recommender "**. we have also declared menu bars which can ease up the complexity of this software up to 10% . codes we used to do this much is provided bellow.

```python
from tkinter import *
import tkinter
from tkinter import messagebox
import datetime
root=Tk(className=" Health Tracker and Recommender")
root.iconbitmap('images/medical.ico')
root.geometry("1080x720")

#Function of menu bar command
#->>>Creating  About us Function of menu bar
frame_root=LabelFrame(root,bg="white")
frame_root.pack()

#creating Whole Window pane
panel_1=PanedWindow(relief="raised")
panel_1.pack(fill=BOTH,expand=True)

#Creating Left Pane screen

left_label=Frame(panel_1,width=100)
panel_1.add(left_label)

#Creating Right and centered Pane
panel_2=Frame(panel_1,relief=FLAT,bg="white")
panel_1.add(panel_2) #Creating Menu Bar:
menu_bar=Menu(root)
root.config(menu=menu_bar)

#creating File Menu
file_menu=Menu(menu_bar)
menu_bar.add_cascade(label="File",menu=file_menu)
file_menu.add_command(label="New File",command=menu_new_file)
file_menu.add_separator()
file_menu.add_command(label="Exit",command=root.destroy)

#creating View Menu
view_menu=Menu(menu_bar)
menu_bar.add_cascade(label="View",menu=view_menu)
view_menu.add_command(label="Enter your Info",command=m_window)
view_menu.add_separator()
view_menu.add_command(label="Check Yourself",command=check_health_alloption)
view_menu.add_command(label="Health Recommender",command=health_recom_fn)

#creating Help Menu
help_menu=Menu(menu_bar)
menu_bar.add_cascade(label="Help",menu=help_menu)
help_menu.add_command(label="New...",command=None)
help_menu.add_separator()
help_menu.add_command(label="About us",command= menu_about)
```

```
#Adding Logo to Left Pane
canvas=Canvas(left_label,width=0,height=0)
canvas.pack(expand=YES,fill=BOTH,padx=20)
gig1=PhotoImage(file='images/Logo.gif')
canvas.create_image(100,150,image=gig1)

#Adding Button to Left Pane
redbutton=Button(left_label,text=" Check
Yourself",relief=RAISED,padx=45,pady=70,font="Algerian",command=check_health_alloption)
redbutton.pack()

button=Button(left_label,text="Health
Recommender",relief=RAISED,padx=20,pady=70,font="Algerian",command=health_recom_fn)
button.pack()

root.resizable(False,False)
root.mainloop()
```
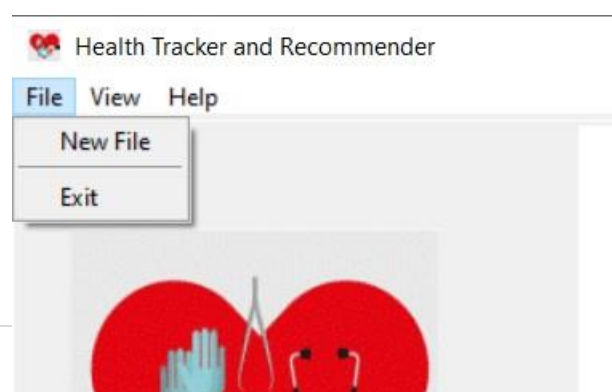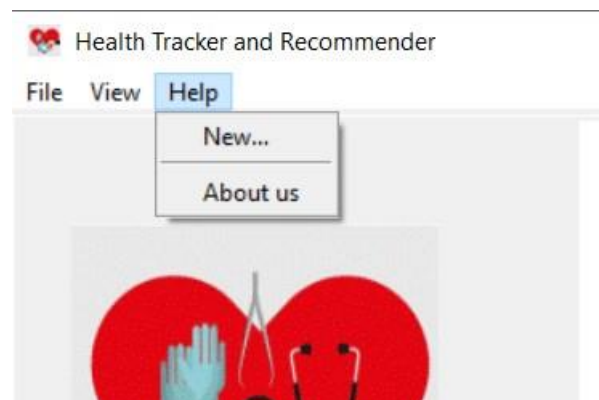
Interface menu –

In menu we are providing 3 menu sections **file, view** and **help**. File is having 2 cascades new file and exit, view is having 3 cascades enter your info , check yourself and health recommender and the final help is having 2 cascades new and about us.

Code we used to declare this much is written bellow

```
#Creating Menu Bar:
menu_bar=Menu(root)
root.config(menu=menu_bar)

#creating File Menu
file_menu=Menu(menu_bar,tearoff=0)
menu_bar.add_cascade(label="File",menu=file_menu)
file_menu.add_command(label="New File",command=menu_new_file)
file_menu.add_separator()
file_menu.add_command(label="Exit",command=root.destroy)

#creating View Menu
view_menu=Menu(menu_bar,tearoff=0)
menu_bar.add_cascade(label="View",menu=view_menu)
view_menu.add_command(label="Enter your Info",command=m_window)
view_menu.add_separator()
view_menu.add_command(label="Check  Yourself",command=check_health_alloption)
view_menu.add_command(label="Health Recommender",command=health_recom_fn)

#creating Help Menu
help_menu=Menu(menu_bar,tearoff=0)
menu_bar.add_cascade(label="Help",menu=help_menu)
help_menu.add_command(label="New...",command=None)
help_menu.add_separator()
help_menu.add_command(label="About us",command= menu_about)
```
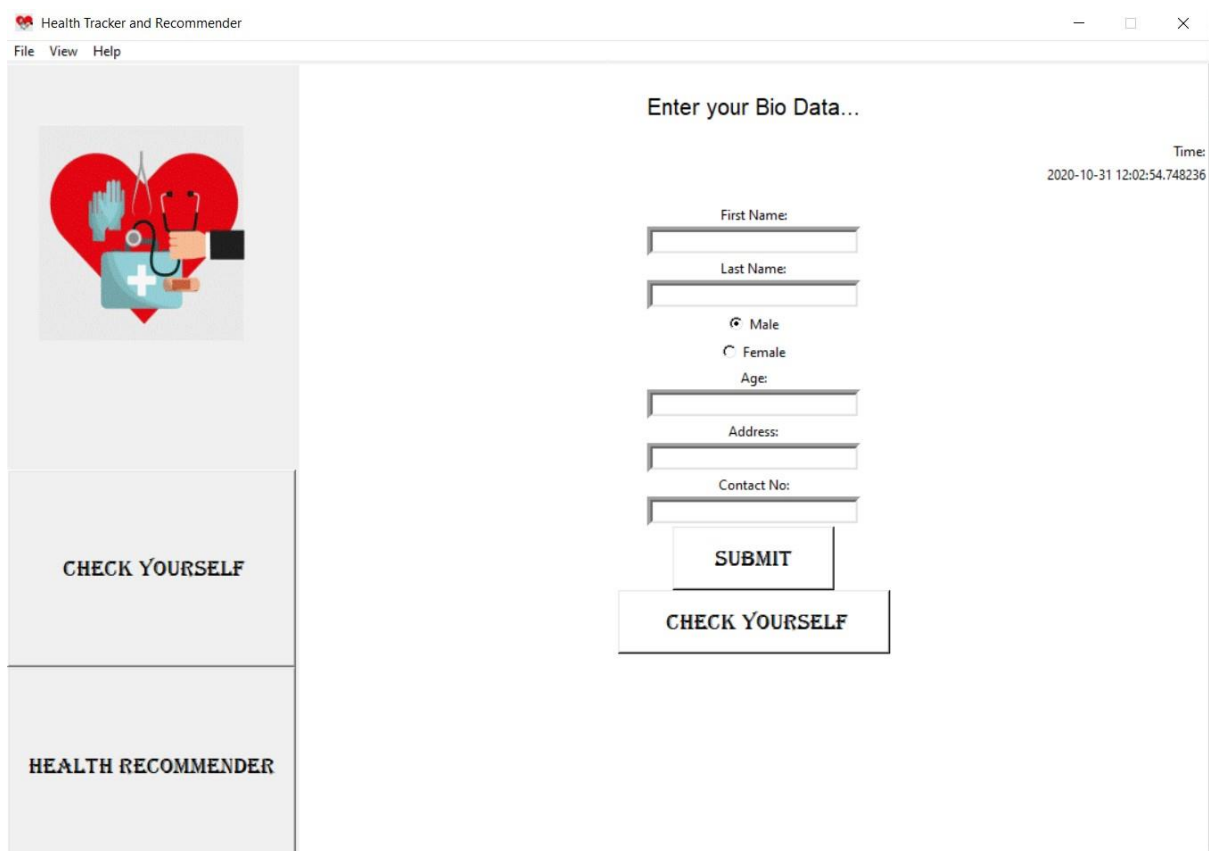
```
#Function of menu bar command
#->>>Creating  About us Function of menu bar
def menu_about():
response=messagebox.showinfo("About us","Developed by : Anurag Anand, Sushant
Kumar, Pabitra Panda \nContact us:  \nE-mail: track.health@gmail.com \nPhone: +91
1800125690  \nGet more info at www.TrackHealth.org")


def menu_new_file():
        for widget in panel_2.winfo_children():
        widget.destroy()
        check_button=Button(panel_2,text="Click here to Enter your
        Information",bg="white",padx=10,pady=10,command=m_window)
        check_button.pack(anchor=CENTER,pady=10)
```

Click here to enter your information –

This button on the panel_2 is declared to help user to reach them out to the information section where they can enter their personal information like name, gender age, address and contact number. Through this page they can either submit their information which will help this software to generate there health report later or they can directly check there health after entering there info.

Codes to declare this much of bio data section is given bellow

```
button1=Button(m_window_canvas,text="Click here to Enter your
Information",bg="#3FEBD8",font=8,relief="raised",padx=5,pady=5,command=m_window)
button1.pack(side="right",pady=10)

def m_window():
    for widget in panel_2.winfo_children():
        widget.destroy()

    x = datetime.datetime.now()
```

```python
Label(panel_2,text="\nEnter your Bio Data...",font=30,bg="white").pack(anchor=N)
Label(panel_2, text="\nTime:",bg="white").pack(anchor=NE)
Label(panel_2, text=x,bg="white").pack(anchor=NE)

Label(panel_2, text="\nFirst Name:",bg="white").pack()
f_name=Entry(panel_2,border=5,width=30)
f_name.pack()
Label(panel_2, text="Last Name:",bg="white").pack()
l_name=Entry(panel_2,border=5,width=30)
l_name.pack()

var=IntVar()
male=Radiobutton(panel_2,text="Male",bg="white",variable=var,value=True,command=
None)
male.pack()

female=Radiobutton(panel_2,text="Female",bg="white",variable=var,value=False,comma
nd=None)
female.pack()

Label(panel_2, text="Age:",bg="white").pack()
age=Entry(panel_2,border=5,width=30)
age.pack()
Label(panel_2, text="Address:",bg="white").pack()
address=Entry(panel_2,border=5,width=30)
address.pack()
Label(panel_2, text="Contact No:",bg="white").pack()
contact=Entry(panel_2,border=5,width=30)
contact.pack()

#submit button
submit_button=Button(panel_2,text="Submit",relief=RAISED,padx=30,pady=10,font="Alg
erian",bg="white",command=submit_fn)
submit_button.pack()
#check yourself button
check_button=Button(panel_2,text=" Check
Yourself",relief=RAISED,bg="white",padx=30,pady=10,font="Algerian",command=check_h
ealth_alloption)
check_button.pack()

m_window_canvas=Canvas(panel_2,height=600,width=500,bg="white",relief=RAISED)
m_window_canvas.pack(expand=YES,fill=BOTH,side="left")
gig=PhotoImage(file='images/apple_.gif')
button1=Button(m_window_canvas,text="Click here to Enter your
Information",bg="#3FEBD8",font=8,relief="raised",padx=5,pady=5,command=m_window)
button1.pack(side="right",pady=10)
m_window_canvas.create_image(300,300,image=gig)
```

Check yourself –

This section is the most important part of this software. This section is designed in such a way so that the user can get their proper current health status. And after which they can go to the health recommender system where they can check that what they have to do next to maintain a good health. This section tells users about their body mass index, blood pressure and sugar level status and generates the final report with current body mass index, BMI status, sugar status, blood pressure status

Code required to declare Check yourself section

```python
redbutton=Button(left_label,text=" Check
Yourself",relief=RAISED,padx=45,pady=70,font="Algerian",command=check_health_allopt
ion)
redbutton.pack()

def check_health_alloption():
   for widget in panel_2.winfo_children():
      widget.destroy()

   Label(panel_2,text="Track your Health
Status...",font=("Algerian",20),bg="white").pack(anchor=N)

   #canvas inside panel_2 for report generatioin
   global check_health_canvas
   check_health_canvas=Canvas(panel_2,bg="white",width=500,height=400)
   check_health_canvas.pack(fill=BOTH,side="right")

   #BMI
   global height
   Label(panel_2,text="BMI Calulator",font=18,bg="white").pack(anchor=NW)
   Label(panel_2, text="\nHeight (metre):",bg="white").pack(anchor=NW)
   height = Entry(panel_2,border=5,width=30)
   height.pack(anchor=NW)
   global weight
   Label(panel_2, text="Weight (Kg):",bg="white").pack(anchor=NW)
   weight =Entry(panel_2,border=5,width=30)
   weight.pack(anchor=NW)

   #Sugar Level

   global sugar_level
   Label(panel_2,text="\nSugar Level Calulator",font=18,bg="white").pack(anchor=NW)
   Label(panel_2, text="\nSugar Level:",bg="white").pack(anchor=NW)
   sugar_level=Entry(panel_2,border=5,width=30)
   sugar_level.pack(anchor=NW)

   #Blood Pressure

   global bp_high

   Label(panel_2,text="\nBlood Pressure
Calulator",font=18,bg="white").pack(anchor=NW)
   Label(panel_2,text="\nBP(Upper):",bg="white").pack(anchor=NW)
   bp_high=Entry(panel_2,border=5,width=30)
   bp_high.pack(anchor=NW)
```

```python
    global bp_low
    Label(panel_2,text="BP(Lower)",bg="white").pack(anchor=NW)
    bp_low=Entry(panel_2,border=5,width=30)
    bp_low.pack(anchor=NW)


    #Repport Generate button
    button1=Button(panel_2,text="Generate
report",bg="#3FEBD8",padx=10,pady=10,command=check_health)
    button1.pack(anchor=SW,pady=10)
# Check Yourself Button Command
def check_health():
    #BMI calculation function
    a=float(height.get())
    b=float(weight.get())
    c=float(a * b)
    bmi = b/(a*a)
    Label(check_health_canvas,text ="Generated
Report\n",font=20,bg="white").pack(side="top")
    Label2=Label(check_health_canvas)
    if ( bmi < 16):
        Label2.configure(text = "\nPhysical Status: Severely
Underweight",font=18,bg="white")

    elif ( bmi >= 16 and bmi < 18.5):
        Label2.configure(text ="\nPhysical Status: Underweight",font=18,bg="white")

    elif ( bmi >= 18.5 and bmi < 25):
        Label2.configure(text ="\nPhysical Status: Healthy",font=18,bg="white")

    elif ( bmi >= 25 and bmi < 30):
        Label2.configure(text = "\nPhysical Status: Overweight",font=18,bg="white")

    elif ( bmi >=30):
        Label2.configure(text ="\nPhysical Status:Severely Overweight",font=18,bg="white")

    Label1=Label(check_health_canvas)
    b=('BMI Index')
    a=(b,bmi)
    Label1.configure(text = a,font=18,bg="white")
    Label1.pack(anchor=NW)
    Label2.pack(anchor=NW)

    #Sugar Level Calculation function
    sugar=int(sugar_level.get())
    Label1=Label(check_health_canvas)
    if sugar>100:
```

```python
        Label1.configure(text="\nSugar Level : High.",font=18,bg="white")
    elif sugar<60:
        Label1.configure(text="\nSugar Level : Low.",font=18,bg="white")
    else:
        Label1.configure(text="\nSugar Level : Fine.",font=18,bg="white")
    Label1.pack(anchor=NW)


    #Blood Pressure Level calculation function
    #->>>>>Global check health function for Blood Pressure
    bpl_Label=Label(check_health_canvas)
    def low_bp(up,down):
        bpl_Label.configure(text="\nyou are facing LOW blood
pressure",font=18,bg="white")
        bpl_Label.pack(anchor=NW)


    bpn_Label=Label(check_health_canvas)
    def normal_bp(up,down):
        bpn_Label.configure(text="\nEnjoy you life",font=18,bg="white")
        bpn_Label.pack(anchor=NW)



    def high_bp(up,down):
        bp_high=up
        bp_low=down
        bph_Label=Label(check_health_canvas)

        if((int(bp_high)<130 and int(bp_high)>=120 )and int(bp_low)<=80):
            bph_Label.configure(text="\nyou area in PREHYPERTENTION
STAGE",font=18,bg="white")
        elif((int(bp_high)<140 and int(bp_high)>=130) and int(bp_low)<90):
            bph_Label.configure(text="\nYou are in HYPERTENTION STAGE
1",font=18,bg="white")
        elif((int(bp_high)<180 and int(bp_high)>=140) and int(bp_low)<120):
            bph_Label.configure(text="\nYou are in HYPERTENTION STAGE
2",font=18,bg="white")
        else:
            bph_Label.configure(text="\nYou are in HYPERTENSIVE CRISIS(Final stage of
HYPERTENTIOIN",font=18,bg="white")
        bph_Label.pack(anchor=NW)

    # main BP caller function
    global up
    up=int(bp_high.get())
    global down
    down =int(bp_low.get())
    bp_Label=Label(check_health_canvas)
    if(up<90 and up>=40 and down<60 and down>=40):
```

```
    low_bp(up,down)

elif(up<120 and up>=90 and down<80):
   normal_bp(up,down)

elif(up>=120 and down>=80):
   high_bp(up,down)
else:
   bp_Label.configure(text="\nPlease enter a valid blood pressure",font=18,bg="white")
   bp_Label.pack(anchor=NW)
```
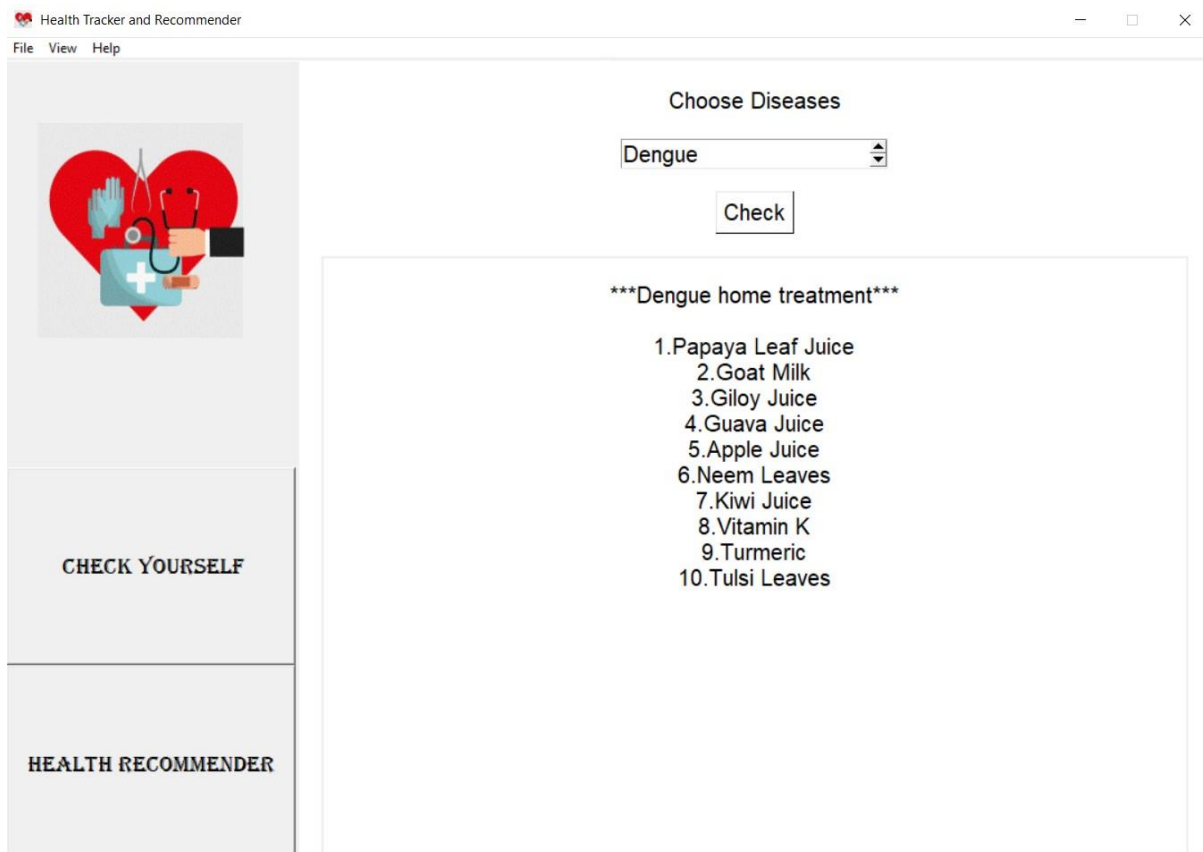
Health Recommender –

This section helps users by providing them the proper guidance about what to do and what not to do in each type of medical condition. This will help to allot users a set of meals, exercises and tips which they need to follow to maintain or heal there medical life. This concept can save the lots of money from the users pocket.

Code require to build this section is given bellow

```python
button=Button(left_label,text="Health
Recommender",relief=RAISED,padx=20,pady=70,font="Algerian",command=health_recom
_fn)
button.pack()
def health_recom_fn():
    for widget in panel_2.winfo_children():
        widget.destroy()

    panel_2.pack_forget()

    w = Label(panel_2, text ='Choose Diseases', font = "50",bg="white")
    w.pack(pady=20)


    global sp
    sp = Spinbox(panel_2,values=("Dengue","Malaria","Corona","Typhoid","Common
Cold","Allergies"),bg="white",fg="Black",font=30)

    sp.pack()
    #calling other_diceases function

disease_compute=Button(panel_2,text="Check",border=1,relief="raised",bg="white",font
=20,command=other_diseases)
    disease_compute.pack(pady=20)
    global cnvs
    cnvs=Canvas(panel_2,width=0,height=0,bg="white")
    cnvs.pack(expand=YES,fill=BOTH,padx=20)

#other_disease_contents
def other_diseases():
    for widget in cnvs.winfo_children():
        widget.destroy()
    Label3=Label(cnvs,text='',bg="white",font=40)
    Label3.pack(pady=20)
    temp=sp.get()
    if(temp=='Dengue'):
        Label3.configure(text="***Dengue home treatment***\n\n1.Papaya Leaf
Juice\n2.Goat Milk\n3.Giloy Juice\n4.Guava Juice\n5.Apple Juice\n6.Neem
Leaves\n7.Kiwi Juice\n8.Vitamin K\n9.Turmeric\n10.Tulsi Leaves")
    elif(temp=="Malaria"):
        Label3.configure(text="***Malaria home treatment***\n\n1.Orange
Juice\n2.Ginger\n3.Lime or lemon Juice\n4.Holy Basil\n5.Warm Water\n6.Grapeffruit
Juice\n7.Berries")
    elif(temp=="Corona"):
        Label3.configure(text="***Corona home treatment***\n\n1.Warm
```

water\n2.Giloy\n3.Tulsi(Basil)\n4.Turmeric powder\n5.Vitamin C rich foods(Kiwi,orange,papaya...)")

    elif(temp=="Typhoid"):

       Label3.configure(text="***Typhoid home treatment***\n\n1.Drink lots of fluids\n2.Garlic\n3. Basil\n4.Apple cider vinegar\n5.Cold compress")

    elif(temp=="Common Cold"):

       Label3.configure(text="***Common Cold home treatment***\n\n1.Garlic\n2.Raw Honey\n3.Ginger\n4.Chicken Soup\n5. Red Onion\n6.Black Pepper\n7.Turmeric Milk")

    else:

       Label3.configure(text="***Allergies home treatment***\n\n1.Use HEPA filters\n2.Butterbur\n3.papaya and pineapple\n4.Honey\n5.Vitamin C")

About us –

This section is available on the menu help bar "About us". These are some basic information about developers team and the contact info
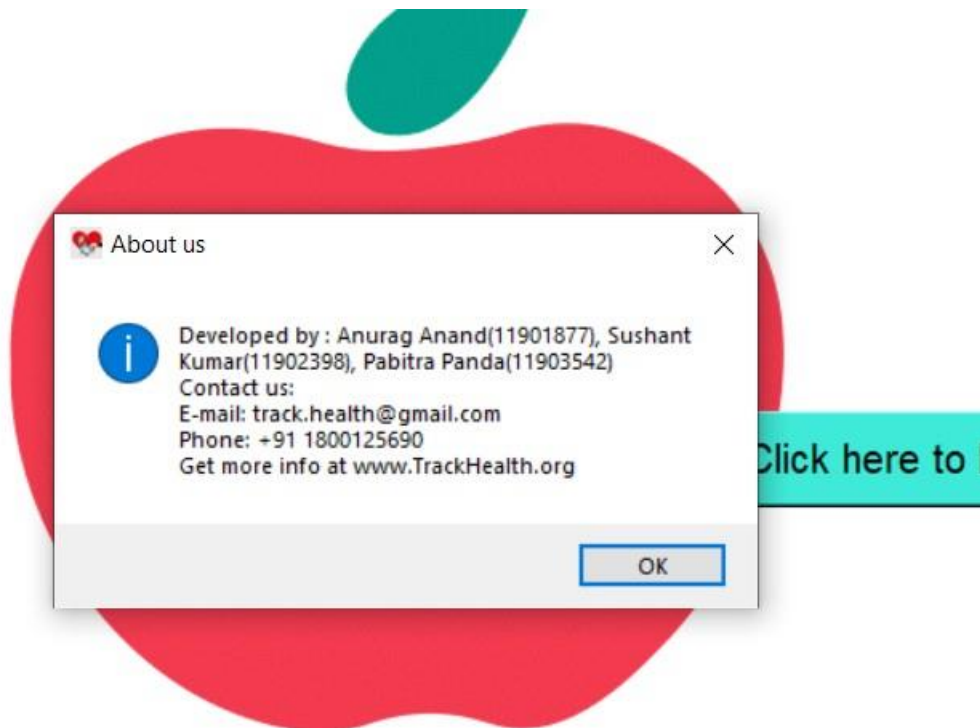
Code required to develop about us section

#declaring menu bar cascade command
help_menu.add_command(label="About us",command= menu_about)

#function to provide the info

```
def menu_about():
    response=messagebox.showinfo("About us","Developed by : Anurag Anand(11901877),
Sushant Kumar(11902398), Pabitra Panda(11903542) \nContact us:  \nE-mail:
track.health@gmail.com \nPhone: +91 1800125690  \nGet more info at
www.TrackHealth.org")
```

ABOUT TECHNOLOGIES AND FRAMEWORK USED

Python, IDLE , Tkinter and its widgets –

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

IDLE is the integrated development environment (IDE) provided with Python. An IDE combines a program editor and a language environment as a convenience to the programmer. Using IDLE is not a requirement for using Python. There are many other IDEs that can be used to write Python programs, not to mention a variety of text-based programmer's editors that many programmers prefer to IDEs.

We had created this GUI using **tkinter module** which is a Python binding to the Tk Gui toolkit. It is the standard Python **interface** to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.
The name *Tkinter* comes from *Tk interface*. Tkinter was written by Fredrik Lundh.

Tkinter is free software released under a Python licence.

We used 2 modules of python in this program. 1st is tkinter which help to create and organize this GUI and 2nd is the datetime module of through which we are fetching the date and the time from the system to show the user. We used different widget defined inside of tkinter interface which provided us the flexibility up to a great extent. Widgets such as label, button, canvas, frame, spinbox, entry, text and menu

WORK DISTRIBUTION

Distribution of roll among students

Anurag Anand –
- Final GUI creation
- Final GUI assembling
- Created functions for several events ,Menu bar Creation and its handling
- Content for Body Mass Index
- Final report framework and About us section
- GitHub Link: https://github.com/01anuraganand/Health_Tracker_and_Recommender.git

Sushant kumar-
- Basic GUI of health recommender
- Final report project description, appendix section and finishing
- Created functions for Blood pressure
- Provided content health recommender and BP
- GitHub Link: https://github.com/000Sushant/health_tracker_and_recommender

Pabitra kumar panda-
- Basic functions for Sugar level
- Provided content health recommender and Sugar level
- Basic GUI for "Click here to enter your information" Button
- Final report Background and Work description section
- GitHub Link: https://github.com/pkpdps2000/Health_Tracker_and_Recommender.git

THANK YOU

By-
ANURAG ANAND
SUSHANT KUMAR and
PABITRA KUMAR PANDA

By-
ANURAG ANAND
SUSHANT KUMAR and
PABITRA KUMAR PANDA

25 | P a g e