AI BASED HEALTH CARE

1. INTRODUCTION

1. OVER VIEW

It is well known that the libraries available in Python for data loading, management, and building models, such as [Pandas](https://www.projectpro.io/article/python-pandas-project-ideas/580), [NumPy](https://www.projectpro.io/data-science-in-python-tutorial/python-numpy-tutorial-for-beginners), and Scikit-Learn, help build robust data science applications. BMI, short for Body Mass Index, is a measure of relative weight based on the mass and height of an individual. We generally use the Body Mass Index in order to categorize people on the basis of their height , blood pressure, ECG, heartbeat rate and weight. A blood pressure reading of 120/80 is considered normal. Heart rate, also called pulse, is the number of times your heart beats per minute. In general, a high pulse or heart rate is more than 100 beats per minute. Other factors may cause variations to this number as well. A BMI Calculator accepts the weight and height of an individual and calculates the Body Mass Index (BMI) of that person. In the above snippet of code, we have defined two variables as the\_height and the weight which uses the input() function to accept input from the user. We have also included the float() function outside the input() function in order to convert the input string into the integer data type so that we can perform calculations with it. To generate report on Tkinter screen.

**2. LITERATURE REVIEW**

* 1. **EXISTING PROBLEM**

Remote health monitoring can provide useful physiological information in the home. This monitoring is useful for elderly or chronically ill patients who would like to avoid a long hospital stay. Wireless sensors are used to collect and transmit signals of interest and a processor is programmed to receive and automatically analyze the sensor signals. In this project, you are to choose appropriate sensors according to what you would like to detect and design algorithms to realize your detection. Examples are the detection of a fall, monitoring cardiac signals. Using a single parameter monitoring system, an approach to a remote health monitoring system was designed that extends healthcare from the traditional clinic or hospital setting to the patient's home. The system was to collect a heartbeat detection system data, fall detection system data, temperature data and few other parameters.

* 1. **PROPOSED SOLUTION**

This system implements an AI based healthcare monitoring system with Ariltifical Health care The system architecture consist of three major parts 1)Heart Beat Rate 2)ECG Reading nodes with a Ariltifical Intelligence and 3)Blood Pressure. In this project the proposed method uses artificial neural network as a gateway to communicate with various sensors such as height, heart rate, and ECG, bp and heartbeat. The sensors are made small and efficient so that their energy storage can last for long time. The proposed system is differ from other Implemented ones because the monitoring of several parameters from many patients simultaneously represents a real time implementation in hospital environments

**3. THERITICAL ANALYSIS**

**3.1 BLOCK DIAGRAM**

Home Page

Patient Personal Data

1. Age
2. Height
3. weight
4. Blood Pressure
5. Heart rate
6. ECG Reading

(Support Vector Machine)

Prediction of AI

Result from Disease

Dataset Entry

3.2 HARDWARE /SOFTWARE DESIGNING

**3.2.1 HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**3.2.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Front-End : Python
* Backend : MYSQL

**SOFTWARE DESIGN**

**PYTHON:**

In technical terms, Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options. Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers. Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects. Once you've developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules. One of the most promising benefits of Python is that both the standard library and the interpreter are available free of charge, in both binary and source form. There is no exclusivity either, as Python and all the necessary tools are available on all major platforms. Therefore, it is an enticing option for developers who don't want to worry about paying high development costs. If this description of Python over your head, don't worry. You'll understand it soon enough. What you need to take away from this section is that Python is a programming language used to develop software on the web and in app form, including mobile. It's relatively easy to learn, and the necessary tools are available to all free of charge. That makes Python accessible to almost anyone. If you have the time to learn, you can create some amazing things with the language

TKINTER:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.  
**To create a tkinter app:**

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

Tkinter messagebox:

The messagebox module is used to display the message boxes in the python applications. There are the various functions which are used to display the relevant messages depending upon the application requirements.

**4. EXPERIMENTAL INVESTIGATIONS**

This approach works with any Python. In the first module, we will collect the data required from the user after getting the details from the Python for the dataset used in Python-based machine learning. The second module will estimate the outcome, or the user's condition throughout the day, by using information obtained in the initial module. In the third module, they will check for abnormality and do by the instructions and conditions given in the architectural flow. IN Fourth Module, the user will be presented with the report produced by the second module, and it will also be stored in the user's phone database so that they can access the data whenever they wish. This idea is developed in such a way that anyone of any age can access it easily.

**5.FLOW CHART**

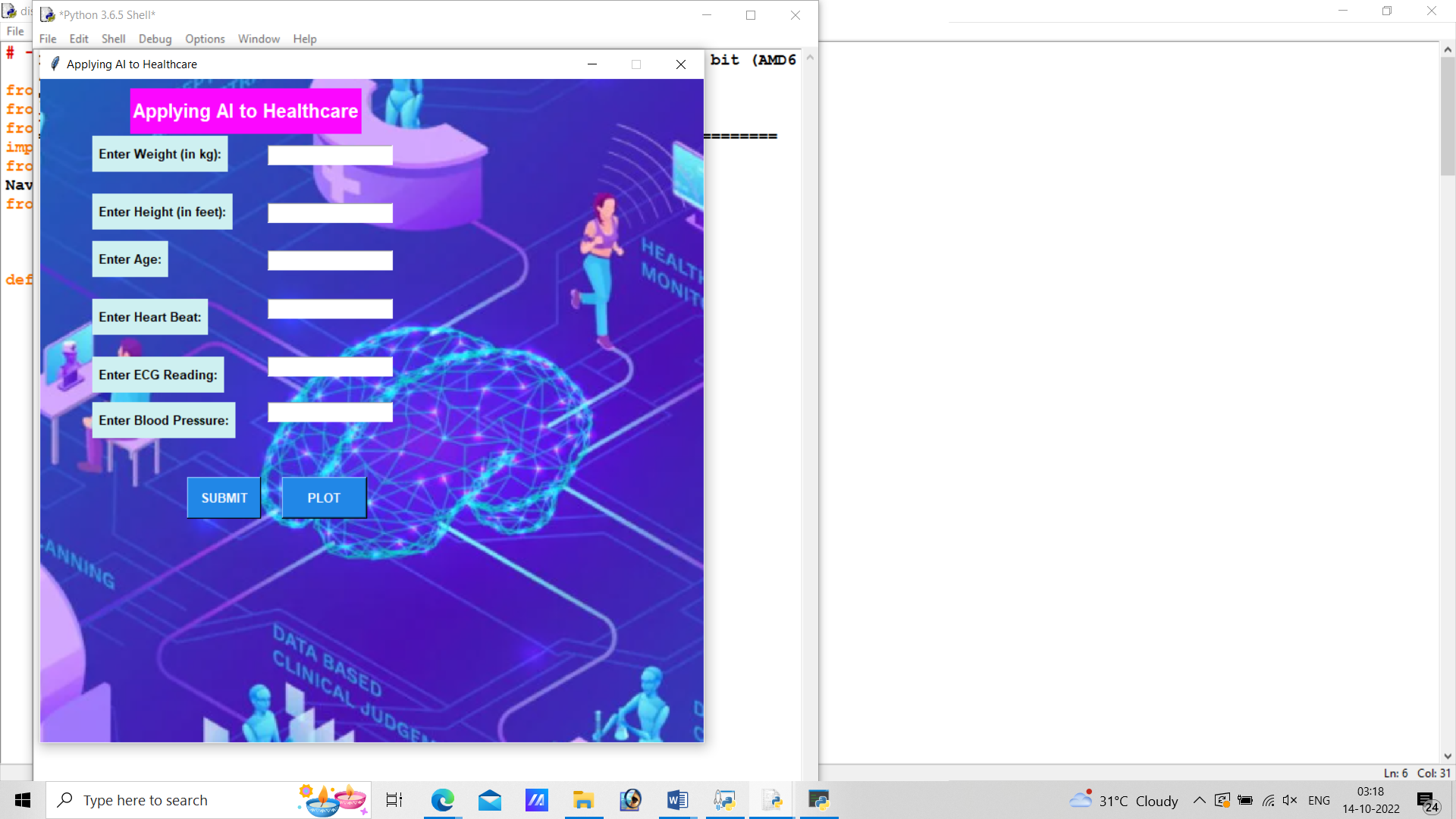
Ask user to enter the height, weight, blood pressure, age, Ecg level, pulserate using tkinter

Abnormal values measured

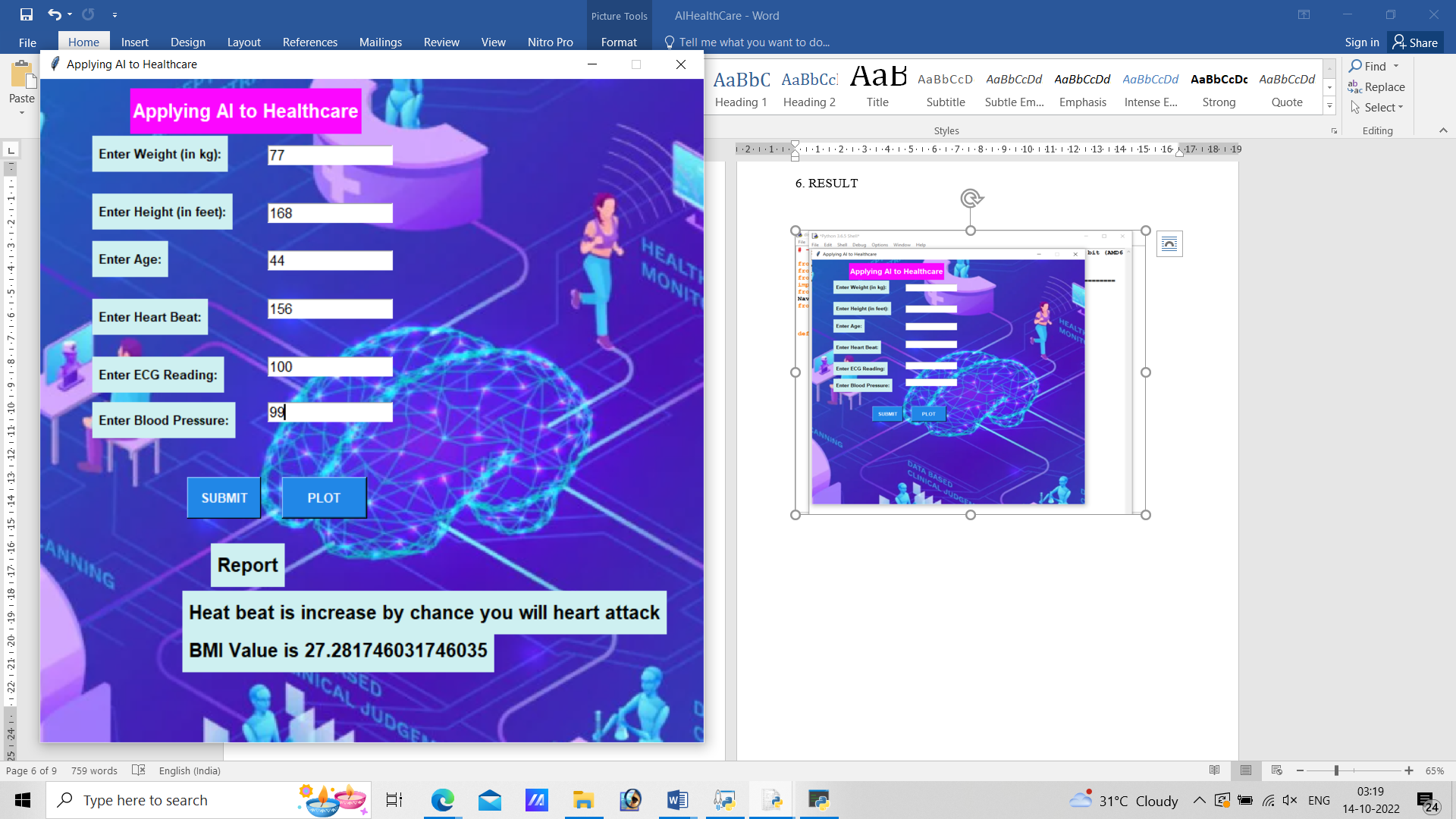
Bmi and possibility disease are displayed in window

**6.RESULT**

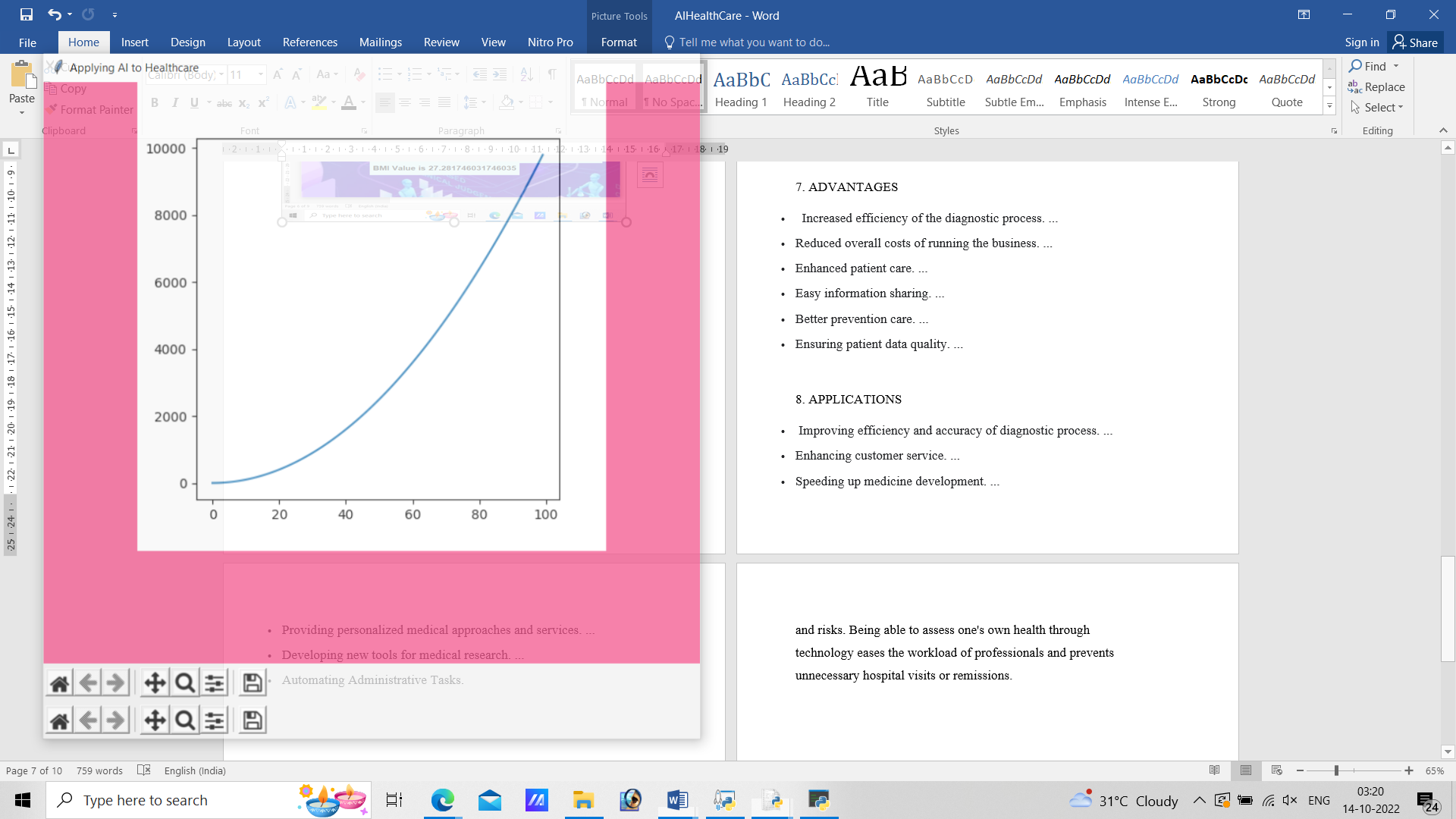
Enter The Data



Result Prediction



Graph



**7. ADVANTAGES**

ADVANTAGES

* Increased efficiency of the diagnostic process. ...
* Reduced overall costs of running the business. ...
* Enhanced patient care. ...
* Easy information sharing. ...
* Better prevention care. ...
* Ensuring patient data quality. …

**DISADVANTAGES**

DISADVANTAGES

* Since We use TKinder as our base we cannot import dataset values in this code…
* We need to rerun our program to clear the previous output ran by the user…

**8. APPLICATIONS**

* Improving efficiency and accuracy of diagnostic process.
* Enhancing customer service.
* Speeding up medicine development.
* Providing personalized medical approaches and services.
* Developing new tools for medical research.
* Automating Administrative Tasks.

**9.CONCLUSION**

In the first method, we will collect the data required from the user after getting the details from the tkinter for the data used in Python-based machine learning. The second method will estimate the outcome, or the user's condition throughout the day, by using information obtained ,they will check for abnormality and do by the instructions and conditions given in the architectural flow. The user will be presented with the report produced by the second method, and it will also be stored in the user's database so that they can access the data whenever they wish. This idea is developed in such a way that anyone of any age can access it easily.

**10.FUTURE SCOPE**

Software that uses AI, like FitBits and smartwatches, can analyze data to alert users and their healthcare professionals on potential health issues and risks. Being able to assess one's own health through technology eases the workload of professionals and prevents unnecessary hospital visits or remissions.

**11.BIBILOGRAPHY**

1. Reddy, S., Fox, J., & Purohit, M. P. (2019). Artificial intelligence-enabled healthcare delivery. Journal of the Royal Society of Medicine, 112(1), 22-28.

2. MailMyStatements, 6 Major Challenges Facing the Healthcare Industry in 2020, Nov 15, 2018 derived from:https://medium.com/@MailMyStatement/5-major-challenges-facing-the-healthcare-industry-in-2019-60218336385f

3. Rita Sharma, Top 10 Challenges Healthcare Companies Face Today, derived from: https://www.finoit.com/blog/top-10-healthcare-challenges/

4. Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. Future healthcare journal, 6(2), 94.

5. Meskó, B., Hetényi, G., & Győrffy, Z. (2018). Will artificial intelligence solve the human resource crisis in healthcare?. BMC health services research, 18(1), 545

6. Lin, S. Y., Mahoney, M. R., & Sinsky, C. A. (2019). Ten ways artificial intelligence will transform primary care. Journal of general internal medicine, 34(8), 1626-1630.

7. Ganapathy, K., Abdul, S. S., & Nursetyo, A. A. (2018). Artificial intelligence in neu-rosciences: A clinician's perspective. Neurology India, 66(4), 934.

8. Etienne, H., Hamdi, S., Le Roux, M., Camuset, J., Khalife-Hocquemiller, T., Giol, M., ... & Assouad, J. (2020). Artificial intelligence in thoracic surgery: past, present, perspective and limits. European Respiratory Review, 29(157).

9. Nadikattu, R. R. (2017). Artificial Intelligence in Cardiac Management. International Journal of Creative Research Thoughts, 5(3).

10. Robert, N. (2019). How artificial intelligence is changing nursing. Nursing manage-ment, 50(9), 30-39.

**APPENDIX**

**A.SOURCE CODE**:

from tkinter import \*

from tkinter import messagebox

from matplotlib.figure import Figure

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import (FigureCanvasTkAgg,

NavigationToolbar2Tk)

from PIL import Image

def plot():

# the figure that will contain the plot

fig = Figure(figsize = (5, 5),

dpi = 100)

# list of squares

y = [i\*\*2 for i in range(100)]

# adding the subplot

plot1 = fig.add\_subplot(111)

plt.show()

# plotting the graph

plot1.plot(y)

# creating the Tkinter canvas

# containing the Matplotlib figure

canvas = FigureCanvasTkAgg(fig,

master = TOP)

canvas.draw()

# placing the canvas on the Tkinter window

canvas.get\_tk\_widget().pack()

# creating the Matplotlib toolbar

toolbar = NavigationToolbar2Tk(canvas,

TOP) toolbar.update()

# placing the toolbar on the Tkinter window

canvas.get\_tk\_widget().pack()

def get\_height():

'''

This function gets height value from Entry field

''' height = float(ENTRY2.get())

return height

def get\_weight():

'''

This function gets weight value from Entry field

'''

weight = float(ENTRY1.get())

return weight

def get\_age():

age = float(ENTRY3.get())

return age

def get\_hb():

hb = float(ENTRY4.get())

return hb

def get\_ecg():

ecg = float(ENTRY5.get())

return ecg

def get\_bb():

bb = float(ENRY6.get())

return bb

def calculate\_age(a=""): # "a" is there because the bind function gives an argument to the function....

print(a)

'''

This function calculates the result

'''

try:

height = get\_height()

weight = get\_weight()

height = height / 100.0

age = get\_age()

age = get\_age()

age1 = weight / (height\*\*2)

hb = get\_hb()

ecg = get\_ecg()

bb = get\_bb()

print(age1)

except ZeroDivisionError:

messagebox.showinfo("Result", "Please enter positive height!!")

except ValueError:

messagebox.showinfo("Result", "Please enter valid data!")

else:

if 75 < hb <= 153 and 60 < ecg <= 100 and 80 < bb <= 130:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Normal No Disease", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

elif 154 < hb <= 500 and 60 < ecg <= 100 and 80 < bb <= 130:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Heat beat is increase by chance you will heart attack", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

elif 0 < hb <= 70 and 60 < ecg <= 100 and 80 < bb <= 130:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Heat beat is increase by chance you will heart attack", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

elif 75 < hb <= 153 and 101 < ecg <= 200 and 80 < bb <= 130:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Abnormal Ecg Reading(High) & Risk", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

elif 75 < hb <= 153 and 0 < ecg <= 59 and 80 < bb <= 130:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Abnormal Ecg Reading(Low)", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

elif 75 < hb <= 153 and 60 < ecg <= 100 and 130 < bb <= 300:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Blood Pressure Level is Higher than Normal & Need Immediate Medical Assetment", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

elif 75 < hb <= 153 and 60 < ecg <= 100 and 0 < bb <= 79:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="Blood Pressure Level is Lower than Normal & Need Immediate Medical Assetment", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

else:

LABLE4 = Label(TOP, bg="#cef0f1", text="Report", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=180, y=490)

LABLE4 = Label(TOP, bg="#cef0f1", text="No Disease", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=540)

LABLE4 = Label(TOP, bg="#cef0f1", text="BMI Value is "+str(age1)+"", bd=6,font=("Helvetica", 15, "bold"), pady=5)

LABLE4.place(x=150, y=580)

f \_\_name\_\_ == '\_\_main\_\_':

TOP = Tk()

TOP.bind("<Return>", calculate\_age)

TOP.geometry("700x700")

TOP.configure(background="#F3578E")

TOP.title("Applying AI to Healthcare")

TOP.resizable(width=False, height=False)

#image=PhotoImage(file="D:\\HealthCarePython\\bank.png")

#l=Label(TOP,image=image)

#l.pack(fill=BOTH)

LABLE = Label(TOP, bg="#FB07FF", text="Applying AI to Healthcare", font=("Helvetica", 15, "bold"), pady=10 ,foreground="white")

LABLE.place(x=95, y=10)

LABLE1 = Label(TOP, bg="#cef0f1", text="Enter Weight (in kg):", bd=6,

font=("Helvetica", 10, "bold"), pady=5)

LABLE1.place(x=55, y=60)

ENTRY1 = Entry(TOP, bd=1, width=16, font="Roboto 11")

ENTRY1.place(x=240, y=70)

LABLE2 = Label(TOP, bg="#cef0f1", text="Enter Height (in feet):", bd=6,

font=("Helvetica", 10, "bold"), pady=5)

LABLE2.place(x=55, y=121)

LABLE3 = Label(TOP, bg="#cef0f1", text="Enter Age:", bd=6,

font=("Helvetica", 10, "bold"), pady=5)

LABLE3.place(x=55, y=171)

LABLE1 = Label(TOP, bg="#cef0f1", text="Enter Heart Beat:", bd=6,

font=("Helvetica", 10, "bold"), pady=5)

LABLE1.place(x=55, y=232)

LABLE2 = Label(TOP, bg="#cef0f1", text="Enter ECG Reading:", bd=6,

font=("Helvetica", 10, "bold"), pady=5)

LABLE2.place(x=55, y=293)

LABLE3 = Label(TOP, bg="#cef0f1", text="Enter Blood Pressure:", bd=6,

font=("Helvetica", 10, "bold"), pady=5)

LABLE3.place(x=55, y=341)

ENTRY2 = Entry(TOP, bd=1, width=16, font="Roboto 11")

ENTRY2.place(x=240, y=131)

ENTRY3 = Entry(TOP, bd=1, width=16, font="Roboto 11")

ENTRY3.place(x=240, y=181)

ENTRY4 = Entry(TOP, bd=1, width=16, font="Roboto 11")

ENTRY4.place(x=240, y=232)

ENTRY5 = Entry(TOP, bd=1, width=16, font="Roboto 11")

ENTRY5.place(x=240, y=293)

ENTRY6 = Entry(TOP, bd=1, width=16, font="Roboto 11")

ENTRY6.place(x=240, y=341)

BUTTON = Button(bg="#2187e7", bd=1, text="SUBMIT",padx=10, pady=10, command=calculate\_age,foreground="white",

font=("Helvetica", 10, "bold"))

BUTTON.place(x=155, y=420)

plot\_button = Button(master = TOP,

command = plot,

height = 2,

width = 10,

text = "PLOT",bg="#2187e7",foreground="white",font=("Helvetica", 10, "bold"))

# in main window

plot\_button.place(x=255, y=420)

#plt.show()

TOP.mainloop()