# BMI Calculator Using Python Report

**Introduction:**

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, resulting from mass in kilograms and height in meters .

**Abstract:**

The BMI Calculator App is a software application which avoids more manual hours that need to spend in personally calculate and find the BMI for a particular person at a single click. This application keeps both the standard in it ie American standard and Indian standard too.This app gives us all the information in both the standards which is not given in existing app.The main scope is to maintain the health. The BMI App gives us all the information ie it gives suggestion for our health and tells us what should we eat and what to avoid. When we enter the height and weight we get all the information ie are we overweight or underweight etc

**Background:**

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 and weight. These categories are underweight, healthy, overweight, and even obesity. Moreover, it is also adopted by various countries in order to promote healthy eating.

We can consider Body Mass Index (BMI) as a substitute for direct measurements of body fat. Besides, BMI is a low-cost and easy-to-perform method of screening for weight classes that may cause health-related problems

A BMI Calculator accepts the weight and height of an individual and calculates the Body Mass Index (BMI) of that person.

For Example, if the height and weight of a person are 155 cm and 57 kg. The BMI of that person will be 23.73 (approx.), which signifies that the person is healthy.

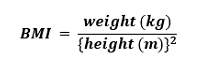
Body Mass Index (BMI) is a measure of body fat on the basis of height and weight, respectively.

On the basis of the BMI of an individual, the calculator returns a statement stating the overall health of the person.

The following table shows how the classification of BMI is done in order to identify the health status of a person.

|  |  |  |
| --- | --- | --- |
| S. No. | BMI | Weight Status |
| **1** | Below 18.5 | Underweight |
| **2** | 18.5 - 24.9 | Normal |
| **3** | 25.0 - 29.9 | Overweight |
| **4** | 30.0 and above | Obese |

We will use the following formula in order to calculate BMI.



Technologies used in this project:

**Python:**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

**Tkinter:**

Tkinter 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 macOS installs of Python. The name Tkinter comes from Tk interface.

**Pycharm IDE:**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

**EXISTING SYSTEM :**

The existing system is somewhat time consuming and hard to calculate. Calculators are small, portable electronic devices used to perform simple as well as complex calculations in afraction of second. The idea of a calculator came from Abacus used long back in 2000 BC; after that there werevarious inventions and mechanical calculators came in use . Scientific calculators were made to solve thescientific calculations. But in the 21st century where people have started using laptops, tablets, smartphones etc.then why would they carry such calculators.The users of this application will have a tool to do all sorts of calculations withease.

**DISADVANTAGES:**

The following are the disadvantages of the existing system -

1. It is very time consuming
2. II. Lot of energy and effort is wasted.
3. III. Wastage of Pages.
4. IV. Hard to maintain old records.
5. Queries are hard apply.

**PROPOSED SYSTEM:**

The BMI Calculator App is software applications which avoids more manual hours that need to spend in personally calculateand find the BMI for a particular person at a single click. This application keeps both the standard in it ie American standard and Indian standard too.This app gives us all the information in both the standards which is not given in existing app.

**SCOPE OF THE SYSTEM:**

The main scope is to maintain the health.

The BMI App gives us all the information i.e it gives suggestion for our health and tells us what should we eat and what to avoid. When we enter the height and weight we get all the information i.e are we overweight or underweight etc.

**FEASIBILITY STUDY:**

A Feasibility Study’s main goal is to assess the economic viability of the proposed business. The feasibility study needs to answer the question “Does the idea make economic sense?” The study should provide a thorough analysis of the business. The outcome of the feasibility study will indicate whether or not to proceed with the proposed venture. If the result of study is positive then the cooperative can proceed to develop a business plan. A feasibility study should examine three main areas:

-Technical Feasibility

-Economic Feasibility

-Operational Feasibility

HARDWARE REQUIREMENTS

|  |  |
| --- | --- |
| **HARDWARE TOOLS** | **MINIMUM REQUIREMENTS** |
| Processor | i3 or other latest version |
| Hard Disk | >500GB |
| RAM | 4GB |
| Monitor | 1 coloured |
| Mouse | 1 |
| Keyboard | 132 keys |

SOFTWARE REQUIREMENTS

|  |  |
| --- | --- |
| **SOFTWARE REQUIREMENTS** | **MINIMUM REQUIREMENTS** |
| Technology | Python Tkinter |
| Scripting Langauge | Python |
| IDE | Pycharm |

**Coding:**

from tkinter import \*

from tkinter import messagebox

def reset\_entry():

age\_tf.delete(0,'end')

height\_tf.delete(0,'end')

weight\_tf.delete(0,'end')

def calculate\_bmi():

kg = int(weight\_tf.get())

m = int(height\_tf.get())/100

bmi = kg/(m\*m)

bmi = round(bmi, 1)

bmi\_index(bmi)

def bmi\_index(bmi):

if bmi < 18.5:

messagebox.showinfo('bmi-bmicalculator', f'BMI = {bmi} is Underweight')

elif (bmi > 18.5) and (bmi < 24.9):

messagebox.showinfo('bmi-bmicalculator', f'BMI = {bmi} is Normal')

elif (bmi > 24.9) and (bmi < 29.9):

messagebox.showinfo('bmi-bmicalculator', f'BMI = {bmi} is Overweight')

elif (bmi > 29.9):

messagebox.showinfo('bmi-bmicalculator', f'BMI = {bmi} is Obesity')

else:

messagebox.showerror('bmi-bmicalculator', 'something went wrong!')

ws = Tk()

ws.title('bmicalculator')

ws.geometry('400x300')

ws.config(bg='#686e70')

var = IntVar()

frame = Frame(

ws,

padx=10,

pady=10

)

frame.pack(expand=True)

age\_lb = Label(

frame,

text="Enter Age (2 - 120)"

)

age\_lb.grid(row=1, column=1)

age\_tf = Entry(

frame,

)

age\_tf.grid(row=1, column=2, pady=5)

gen\_lb = Label(

frame,

text='Select Gender'

)

gen\_lb.grid(row=2, column=1)

frame2 = Frame(

frame

)

frame2.grid(row=2, column=2, pady=5)

male\_rb = Radiobutton(

frame2,

text = 'Male',

variable = var,

value = 1

)

male\_rb.pack(side=LEFT)

female\_rb = Radiobutton(

frame2,

text = 'Female',

variable = var,

value = 2

)

female\_rb.pack(side=RIGHT)

height\_lb = Label(

frame,

text="Enter Height (cm) "

)

height\_lb.grid(row=3, column=1)

weight\_lb = Label(

frame,

text="Enter Weight (kg) ",

)

weight\_lb.grid(row=4, column=1)

height\_tf = Entry(

frame,

)

height\_tf.grid(row=3, column=2, pady=5)

weight\_tf = Entry(

frame,

)

weight\_tf.grid(row=4, column=2, pady=5)

frame3 = Frame(

frame

)

frame3.grid(row=5, columnspan=3, pady=10)

cal\_btn = Button(

frame3,

text='Calculate',

command=calculate\_bmi

)

cal\_btn.pack(side=LEFT)

reset\_btn = Button(

frame3,

text='Reset',

command=reset\_entry

)

reset\_btn.pack(side=LEFT)

exit\_btn = Button(

frame3,

text='Exit',

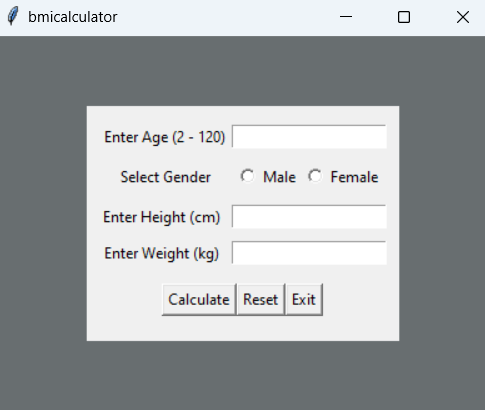
command=lambda:ws.destroy()

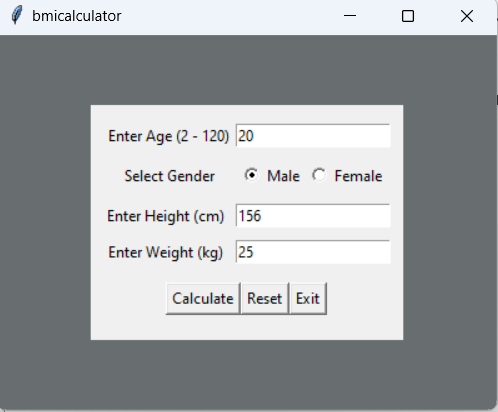
)

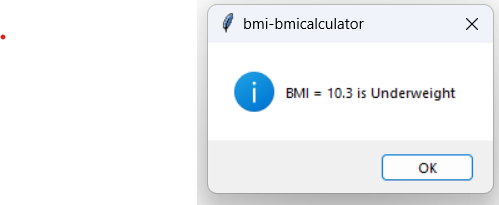
exit\_btn.pack(side=RIGHT)

ws.mainloop()

**Output Screenshots:**

****

****

****

**Future Scope:**

This application avoids the manual work and the problems concern with it. Centralized management of the database &one app to manage the BMI Calculator of the different section of the female/male etc.Well I and my team member have worked hard in order to present an improved project/app better than the existing one’s regarding the information about the various activities. Still, we found out that the project can be done in a better way. We can add alter message to her/him to eat and excise.

**Conclusion:**

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project. Automation of the entire system improves the efficiency It provides a friendly graphical user interface which proves to be better when compared to the existing system. It gives appropriate access to the authorized users depending on their permissions.

 It effectively overcomes the time complexity.  Updating of information becomes so easier.  System security, data security and reliability are the striking features.  The System has adequate scope for modification in future if it is necessary.

**References and Bibliography:**

1. Python Programming ( <https://docs.python.org/3/> )
2. Tkinter ( <https://docs.python.org/3/library/tk.html> )
3. BMI Calculator ( <https://www.calculator.net/bmi-calculator.html> )