A picture containing text, soup

Description automatically generated

**BGS PUBLIC SCHOOL**

**BGS Health and Education City**

**PROJECT SYNOPSIS**

**BMI CALCULATOR**



**Submitted By:**

**VEDANTH.M**

**CLASS: XII ‘A’, CBSE**

**Submitted To:**

**Mrs. SHUBHA H M**

**PGT, Computer Science**

**Department of Computer Science**

**BGS Health and Education City**

**#67, Dr. Vishnuvardhan Road, Bangalore-60.**

**INDEX**

|  |  |
| --- | --- |
| Sl.No | Topic |
| **1** | Acknowledgement |
| **2** | Certification |
| **3** | Motivation |
| **4** | Objective |
| **5** | Application and Practicality |
| **6** | Software Dependencies |
| **7** | Project Plan |
| **8** | Program Flow |
| **9** | Ideal Output |
| **10** | Areas of Improvement |
| **11** | Bibliography |

**Acknowledgement**

We are extremely grateful to all those who have helped us to put these ideas, well above the level of simplicity and into something concrete and creative.

We would like to express our special thanks of gratitude to our teachers as well as our principal who gave us the golden opportunity to do this wonderful project on the topic Blackjack, which also led us to do a lot of research and due to this, we have gained knowledge about so many new things. Any attempt at any level can ‘t be satisfactorily completed without the support and guidance of our parents and friends.

We would like to thank our parents who helped us a lot in gathering different information, collecting data, and guiding us from time to time in making this project, despite their busy schedules, they gave us different ideas in making this project unique.

**BGS PUBLIC SCHOOL**

**BGS Health and Education City**

A close-up of a coin

Description automatically generated with low confidence

**CERTIFICATE**

This is to certify that **VEDANTH M** - student of class XII has successfully completed this **Computer Science** project titled **“BMI CALCULATOR**” during the academic **2022-23** as per the guidelines issued by the **Central Board of Secondary Education**. **Mrs. Shubha H.M** (P.G.T. Computer Science ) Jai Sri Gurudev BGS PUBLIC SCHOOL BGS Health and Education City, #67, Dr. Vishnuvardhana Road, Bangalore-560060.

Mrs. Shubha H.M

(P.G.T. Computer Science)

**BGS PUBLIC SCHOOL**

**BGS Health and Education City**

A picture containing text, soup

Description automatically generated

**CERTIFICATE**

This is to certify that **VEDANTH M** student of class XI have successfully completed this Computer Science project titled “**BMI CALCULATOR**” during the academic year **2022-23** as per the guidelines issued by the **Central Board of Secondary Education**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER PRINCIPAL**

**\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DATE EXTERNAL EXAMINER**

**Motivation**

We wanted the user to measure their Body Mass Index as easily and interactive as possible, which specifically is fairly significant. We wanted this project to be universally accessible to everyone regardless of where they generally are from in a particularly major way

**Objective**

The project's purpose is to allow the user to input their height and weight, and to calculate their Body Mass Index (BMI). Based on general reference values, this information can indicate whether the user is underweight, normal, slightly overweight or obese.

**Application And Practicality Of The Project**

This project aims to help one to measure their Body Mass Index(BMI) and thus this code can be used in a future project involving much more advanced scopes such as:

1. Being able to calculate more than one Body Mass Indexes at once.
2. Being able to export the calculated BMI data into a CSV file

**Software**

For this project, users are recommended to have the following installed on their system for best results:

1. Python version 3.x.x or higher
2. PyCharm version 2021.x.x or higher

**Project Plan**

The following functions and modules from the Python library are used:

* Defined Functions
* User input and output
* ‘Tkinter(tk)’, ‘Path’, ‘messagebox’, ‘ttk’, ‘mysql.connector’ modules
* Several variables

**Expected Output**

It is desired for the user to use the Body Mass Index calculator without any technical glitches or errors during runtime.

The user is expected to read the first two message boxes, as it contains basic information that the user is required to know and understand about the purpose and the use of calculating Body Mass Index.

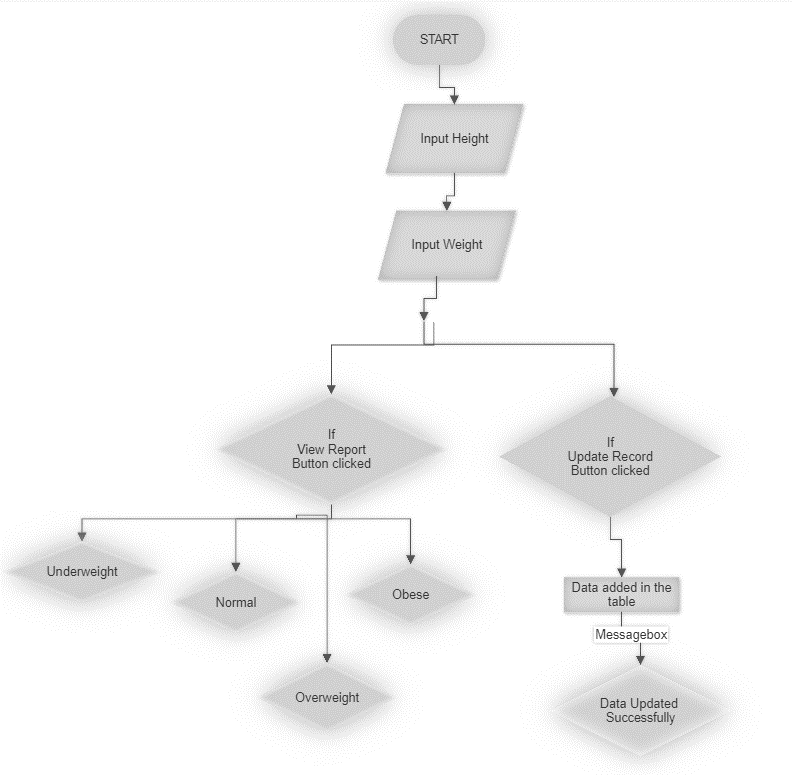
After the user has successfully read the message boxes, the user is then taken to the main window where the user is required to input their height and weight values from the given slider button or can just type it in the entry box.

While measuring the height, the user is also given an interactive image that changes the height of the reference image as the user changes the height value through the slider.

And once the user has finally input their values, the user needs to click on the 'View Report' button to show the results. The user can either get their result as 'Underweight', 'Normal', 'Overweight' and 'Obese', according to the values the user has given in.

The feature with the option, 'Update' also allows the user to update their values in a MySQL database, which gives an advantage to the user to update multiple values into a single table for better and easy reference for the same.

**Program Flow**



**Program Code**

import tkinter as tk

from pathlib import Path

from tkinter import messagebox, ttk

import mysql.connector as pymysql

from PIL import Image, ImageTk

from db\_handler import \*

ASSETS\_PATH = Path(\_\_file\_\_).resolve().parent / "assets"

messagebox.showinfo(

    title="Welcome, User!",

    message="Welcome!\nBody Mass Index (BMI) is widely "

    + "used as an indicator of body fat content.\n",

)

messagebox.showinfo(

    title="Welcome, User!",

    message="If you want to calculate your BMI, you have to find out your "

    + "weight and height first. Once you know these values, "

    + "click OK and proceed",

)

root = tk.Tk()

root.title("BMI Calculator")

root.geometry("470x580+300+200")

root.resizable(False, False)

root.configure(bg="white")

root.focus\_force()

root.bind\_all("<Button-1>", lambda event: event.widget.focus\_set())

getval = ""

# BMI calculation

def BMI():

    global getval

    h = float(Height.get())

    w = float(Weight.get())

    # Convert HEIGHT into METER

    try:

        m = h / 100

        bmi = round(float(w / m\*\*2), 1)

        label1.config(text=bmi)

    except ZeroDivisionError:

        messagebox.showerror(

            title="BMI Calculator", message="Height/Weight cannot be 0!"

        )

    if bmi <= 18.5:

        label2.config(text="Underweight!")

        label3.config(

            text=mycursor\_fetch\_any("bmi\_vals", "bmi\_getter", "name = 'Underweight'")[

                0

            ][0]

        )

        getval = "Underweight"

    elif bmi > 18.5 and bmi <= 25:

        label2.config(text="Normal!")

        label3.config(

            text=mycursor\_fetch\_any("bmi\_vals", "bmi\_getter", "name = 'Normal'")[0][0]

        )

        getval = "Normal"

    elif bmi > 25 and bmi <= 30:

        label2.config(text="Overweight!")

        label3.config(

            text=mycursor\_fetch\_any("bmi\_vals", "bmi\_getter", "name = 'Overweight'")[0][

                0

            ]

        )

        getval = "Overweight"

    else:

        label2.config(text="Obese!")

        label3.config(

            text=mycursor\_fetch\_any("bmi\_vals", "bmi\_getter", "name = 'Obese'")[0][0]

        )

        getval = "Obese"

# ICON

image\_icon = ImageTk.PhotoImage(Image.open(ASSETS\_PATH / "BMIcalcL.png"))

root.iconphoto(False, image\_icon)

# TOP

top = ImageTk.PhotoImage(Image.open(ASSETS\_PATH / "BMIcalc.png"))

top\_image = tk.Label(root, image=top, background="#f0f1f5")

top\_image.place(x=-38, y=-10)

# bottom BOX

tk.Label(root, width=72, height=18, bg="orange").pack(side="bottom")

# TWO BOXES

box = ImageTk.PhotoImage(Image.open(ASSETS\_PATH / "box.png"))

tk.Label(root, image=box).place(x=20, y=100)

tk.Label(root, image=box).place(x=240, y=100)

# SCALE

scale = ImageTk.PhotoImage(Image.open(ASSETS\_PATH / "scale.png"))

tk.Label(root, image=scale, bg="orange").place(x=-10, y=310)

# ###############SLIDER1#####################

current\_value = tk.DoubleVar()

def get\_current\_value():

    return "{: .2f}".format(current\_value.get())

def slider\_changed(event):

    Height.set(get\_current\_value())

    size = int(float(Height.get()))

    img = Image.open(ASSETS\_PATH / "Guy.png")

    resized\_image = img.resize((50, 10+ size))

    photo2 = ImageTk.PhotoImage(resized\_image)

    secondimage.config(image=photo2)

    secondimage.place(x=70, y=540 - size)

    secondimage.image = photo2

# COMMAND TO CHANGE BACKGROUND COLOR OF SCALE

style = ttk.Style()

style.configure("TScale", background="white")

slider = ttk.Scale(

    root,

    from\_=0,

    to=220,

    orient="horizontal",

    style="TScale",

    command=slider\_changed,

    variable=current\_value,

)

slider.place(x=80, y=250)

#############################################

# @@@@@@@@@@@@@@@@@SLIDER2@@@@@@@@@@@@@@@@@@@@@

current\_value2 = tk.DoubleVar()

def get\_current\_value2():

    return "{: .2f}".format(current\_value2.get())

def slider\_changed2(event):

    Weight.set(get\_current\_value2())

def add\_data():

    if float(Height.get()) == 0.00 or float(Weight.get()) == 0.00:

        messagebox.showerror(

            title="BMI Calculator",

            message="Weight/Height cannot be 0!",

        )

    else:

        tup = (BMIval.get(), Height.get(), Weight.get(), getval)

        print(tup)

        bmi\_update(tup)

# COMMAND TO CHANGE BACKGROUND COLOR OF SCALE

style2 = ttk.Style()

style2.configure("TScale", background="white")

slider2 = ttk.Scale(

    root,

    from\_=0,

    to=200,

    orient="horizontal",

    style="TScale",

    command=slider\_changed2,

    variable=current\_value2,

)

slider2.place(x=300, y=250)

# @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

# ENTRY BOX

Height = tk.StringVar()

Weight = tk.StringVar()

BMIval = tk.StringVar(value="")

height = tk.Entry(

    root,

    textvariable=Height,

    width=5,

    font="arial 50",

    bg="#fff",

    fg="#000",

    bd=0,

    justify="center",

)  # 'justify' TO align text in center

height.place(x=35, y=160)

Height.set(get\_current\_value())

weight = tk.Entry(

    root,

    textvariable=Weight,

    width=5,

    font="arial 50",

    bg="#fff",

    fg="#000",

    bd=0,

    justify="center",

)  # 'justify' TO align text in center

weight.place(x=255, y=160)

Weight.set(get\_current\_value2())

name\_entry = tk.Entry(master=root, width=71, textvariable=BMIval, background="#B1ABD4", foreground="#000000", relief = "groove").place(x=20, y=60)

# MAN IMAGE

secondimage = tk.Label(root, bg="orange")

secondimage.place(x=70, y=530)

# CALLING BMI FUNCTION [Line 30] button for bmi calculation

tk.Button(

    root,

    text="View Report",

    width=15,

    height=2,

    font="CalibriBold 10",

    bg="green",

    fg="white",

    command=BMI,

).place(x=320, y=360)

tk.Button(

    root,

    text="Update record",

    width=15,

    height=2,

    font="CalibriBold 10",

    bg="green",

    fg="white",

    command=add\_data,

).place(x=320, y=320)

label1 = tk.Label(root, font="arial 60 bold", bg="orange", fg="#fff")

label1.place(x=125, y=305)

label2 = tk.Label(root, font="arial 20 bold", bg="orange", fg="#3b3a3a")

label2.place(x=280, y=430)

label3 = tk.Label(root, font="arial 10", bg="orange")

label3.place(x=200, y=500)

label4 = tk.Label(root, text="HEIGHT(cm)", font="BahnschriftBold 15", bg="white")

label4.place(x=75, y=115)

label5 = tk.Label(root, text="WEIGHT(kg)", font="BahnschriftBold 15", bg="white")

label5.place(x=285, y=115)

root.mainloop()

**Program Code: (db\_handler)**

import mysql.connector

import tkinter.messagebox

# password = open("password.txt").readline()

mydb = mysql.connector.connect(

    host="localhost", user="root", password="veds", database="bmi"

)

mycursor = mydb.cursor()

def mycursor\_fetch\_any(field, name, cond):

    mycursor.execute(f"select {field} from {name} where {cond}")

    return mycursor.fetchall()

def bmi\_update(tup: tuple):

    mycursor.execute(f"insert into bmi values(%s, %s, %s, %s)", tup)

    mydb.commit()

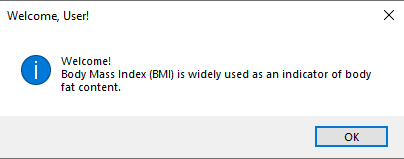
    tkinter.messagebox.showinfo(

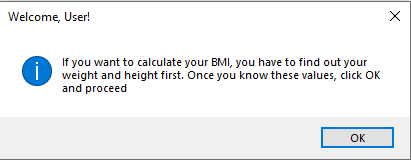
        title="BMI Calculator",

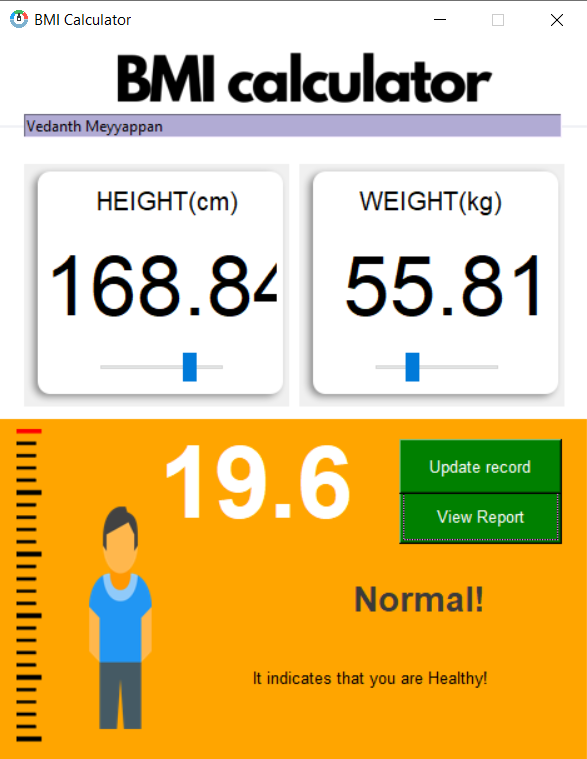
        message="Data inserted successfully.",

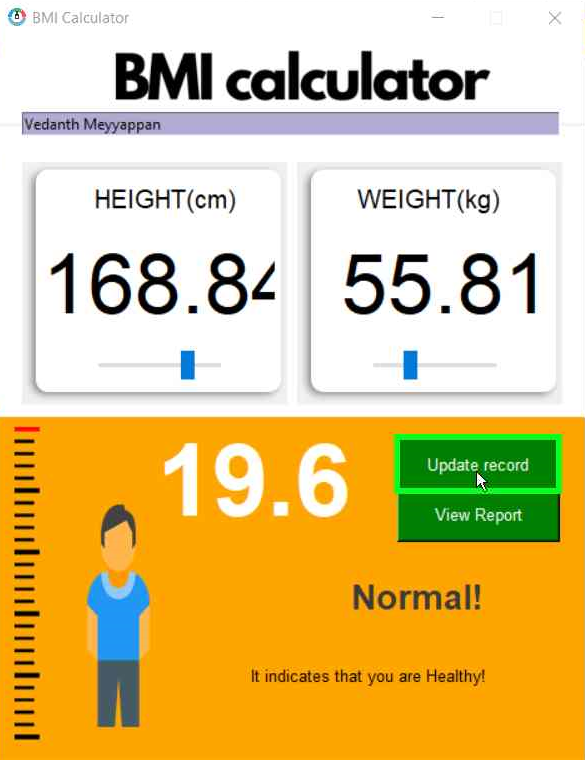
    )

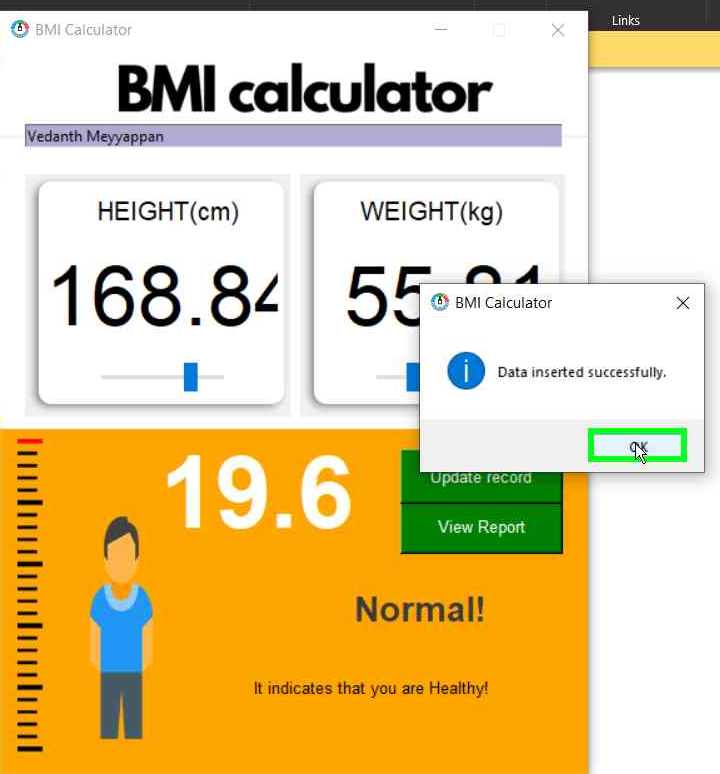
**Output**



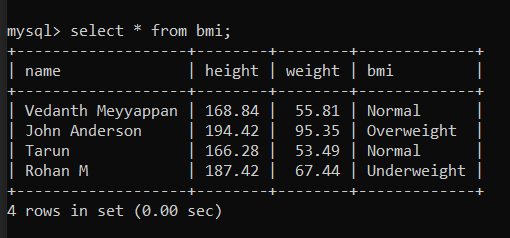








MySQL Command Line Client Output



**Areas of Improvement**

**in the Future**

This project aims to help one to measure their Body Mass Index(BMI) and thus this code can be used in a future project involving much more advanced scopes such as:

1. Being able to calculate more than one Body Mass Indexes at once.
2. Being able to export the calculated BMI data into a CSV file

And etc:-

**Bibliography**

* [Glossary — Python 3.11.0 documentation](https://docs.python.org/3/glossary.html)
* [Reference to BMI calculator code for calculations](https://www.javatpoint.com/create-bmi-calculator-using-python)
* [Reference for Tkinter GUI](https://www.geeksforgeeks.org/python-tkinter-tutorial/)
* Sumita Arora Class XII