Group 8 - J1T: Final Project

BODY MASS INDEX CALCULATOR

Submitted by
Aninang, Kelvin Adam
Cabanes, Angela
Ibanez, Frank
Jornales, Michaela

Submitted to
Prof. Jasmin Almarinez

I. INTRODUCTION

This code provides a user-friendly BMI calculator that allows individuals to input their height and weight. The Tkinter library simplifies the process of creating the graphical elements and handling user interactions, making it a suitable choice for developing basic GUI applications. This paper provides an overview and documentation of the program members of group eight made.

PROBLEM

BMI calculator: This program calculates a user's body mass index (BMI) based on their height and weight inputs, and provides feedback on whether the user is underweight, overweight, or obese.

SOLUTION

The only difficulty in making the code is to make the GUI of the code. There have been many revisions of the said design and parts of the design that were not intended. For example, the members had been struggling to place the tips of each category to not overlap outside of the window. In order to solve this, the members simply add a newline in every part of the text where the text would overlap. Additionally, alignment issues were discovered within the program's infobox. To address the problem, the members implemented a for loop within the infobox to ensure that content would always be centered.

The team encountered challenges in adding a tab for the imperial system, as they were unsure of the appropriate method to incorporate this feature within the tkinter framework. The members did not know how to add one so research was undertaken to identify a suitable solution. It also made another issue since adding the imperial system brings another set of codes such as formulas to be added in the program. The decision to incorporate this feature was motivated by the team's desire to familiarize users with the imperial system, particularly in the United Kingdom. We also wanted to change the look for our button to make it more round
Furthermore, the team sought to enhance the software's graphical user interface by changing the button's appearance to a more rounded shape, along with the appearance of the entry attribute to make the interface more modern looking.

BACKGROUND

Code Components/Techniques:

The weight and height input parameters for the calculate_bmi function are specified in the definition. By dividing the weight (in kilos) by the square of the height (in meters), this function determines the BMI. The calculated BMI is then returned.

The code asks the user to enter their height and weight using the input function in the sample usage section. The string argument that the input function accepts acts as a prompt for the user. The weight and height variables are then given the input values that have been converted to floats using the float function.

Import modules - “tkinter” main function for creating GUI
-”messagebox” displays the error message
-”ttk” notebook style interface
-”customtkinter” custom module that creates a customized tkinter widgets

GUI initialization - ‘customtkinter.CTk() sets window title “BMI CALCULATOR” as well as minimum and maximum dimensions.

Frame and Notebook - top and bottom frame that fixes the GUI layout
Metric Tab - User Information input fields (nickname,height c.m, weight kg)
Imperial tab - User Information input fields (nickname,height ft, weight lbs)

II. PROJECT DESCRIPTION

Using Python/Pycharm, members of group eight created a simple Body Mass Index Calculator and Tkinter for creating An Interactive GUI Window.

This code enables users to input their height and weight then it calculates their Body Mass Index (BMI). The GUI is created using the *tkinter* module and *customtkinter* module to achieve a modern style interface. The program code imports several modules, including *tkinter*, *messagebox*, and *ttk*, to create the GUI. The customtkinter module is used to create customized *tkinter* widgets. It contains features that allow users to choose either metric or imperial system in a tab, depending on the unit the user is accustomed to. The program is then used to provide feedback to the user on their weight status, such as if they are underweight, normal, overweight, or obese along with some informative tips on how to improve their health. The user can also enter their name, although it is an optional feature, that would be displayed with the feedback too. Additionally, when the user inputs invalid or an empty digit inside the height and weight, the system will pop another window that indicates the error in the input.

III. PROGRAM CODE

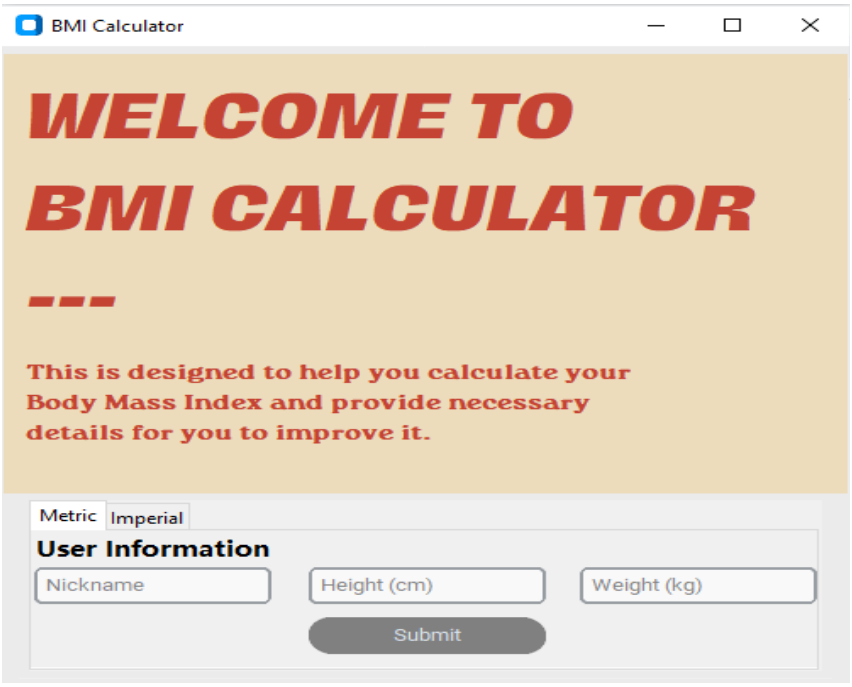
```
import tkinter
from tkinter import messagebox
from tkinter import *
from tkinter import ttk
import customtkinter

def calculate_bmi_m():
    # Check if height and weight fields are not empty
    if height_entry.get() == "" or weight_entry.get() == "":
        messagebox.showerror("Error", "Please enter a value for height and weight.")
        return

    # Check if height and weight fields are valid numbers
    try:
        height = float(height_entry.get()) / 100
        weight = float(weight_entry.get())
        bmi = round(weight / (height * height), 2)
    except ValueError:
        messagebox.showerror("Error", "Please enter a valid number for height and weight.")
        return

    if bmi < 18.5:
        feedback_label.config(text=f"You're underweight\n\nTo improve health, add healthy calories with\nalmonds, "
                              f"fruits, and wheat toast. Choose\nnutrient-dense foods avoid junk food, snack\non "
                              f" fats, and hit the gym for weight-\nlifting and yoga. Prioritize getting enough\n"
                              f"sleep.\n\n{nickname_entry.get()}'s BMI score is " + str(bmi),
                         font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)
    elif bmi < 25:
        feedback_label.config(text=f"You're normal weight\n\nMaintain your healthy weight\n"
                              f"\n{nickname_entry.get()}'s BMI score is " + str(bmi),
                         font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)
    elif bmi < 30:
        feedback_label.config(text=f"You're overweight\n\nTo improve your diet, add fiber-rich green\nleafy vegetables,"
                              f" while avoiding sugar and\nrefined carbohydrates. Regular exercise, like\nwalking "
                              f"or jogging, is beneficial, and staying\nhydrated is important for overall health."
                              f"\n\n{nickname_entry.get()}'s BMI score is " + str(bmi),
                         font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)
    else:
        feedback_label.config(text=f"You're obese\n\nObesity is unhealthy and can lead to heart\nstroke and other "
                              f"health complications. Seek\ndoctor to know more about your current\nhealth\n"
                              f"\n{nickname_entry.get()}'s BMI score is " + str(bmi),
                         font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)

def calculate_bmi_i():
    # Check if height and weight fields are not empty
    if height_entry2.get() == "" or weight_entry2.get() == "":
        messagebox.showerror("Error", "Please enter a value for height and weight.")
        return
    # Check if height and weight fields are valid numbers
    try:
        height = float(height_entry2.get()) * 12 + float(inches_entry.get())
        weight = float(weight_entry2.get())
```

```
        bmi = round(weight / (height * height) * 703, 2)
    except ValueError:
        messagebox.showerror("Error", "Please enter a valid number for height and weight.")
        return

    if bmi < 18.5:
        feedback_label.config(text=f"You're underweight\n\nTo improve health, add healthy calories with\nalmonds, "
                              f"fruits, and wheat toast. Choose\nnutrient-dense foods avoid junk food, snack\non "
                              f"good fats, and hit the gym for weight-\nlifting and yoga. Prioritize getting "
                              f"enough\nsleep.\n\n{nickname_entry2.get()}'s BMI score is " + str(bmi),
                              font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)
    elif bmi < 25:
        feedback_label.config(text=f"You're normal weight\n\nMaintain your healthy weight\n"
                              f"n{nickname_entry2.get()}'s BMI score is " + str(bmi),
                              font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)
    elif bmi < 30:
        feedback_label.config(text=f"You're overweight\n\nTo improve your diet, add fiber-rich green\nleafy vegetables,"
                              f" while avoiding sugar and\nrefined carbohydrates. Regular exercise, like\nwalking "
                              f"or jogging, is beneficial, and staying\nhydrated is important for overall health."
                              f"\n\n{nickname_entry2.get()}'s BMI score is " + str(bmi),
                              font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)
    else:
        feedback_label.config(text=f"You're obese\n\nObesity is unhealthy and can lead to heart\nstroke and other "
                              f"health complications. Seek\ndoctor to know more about your current\nhealth\n"
                              f"\n{nickname_entry2.get()}'s BMI score is " + str(bmi),
                              font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", anchor=W)

window = customtkinter.CTk()
window.title("BMI Calculator")
window.minsize(width=500, height=1)
window.maxsize(width=500, height=1000)

frame = tkinter.Frame(window)
frame.pack()

# create frames
top_frame = tkinter.Frame(window, bg="#ecdcbc")
bottom_frame = tkinter.Frame(window)

# place frames in the window
top_frame.pack(side="top", fill="both", expand=True, padx=1, pady=(5, 5))
bottom_frame.pack(side="bottom", fill="both", expand=True, padx=10, pady=(5, 5))

# notebook

notebook = ttk.Notebook(window)
tab1 = Frame(notebook) # frame for tab 1
tab2 = Frame(notebook)

notebook.add(tab1, text="Metric")
notebook.add(tab2, text="Imperial")
notebook.pack()

# User information in Metric
```

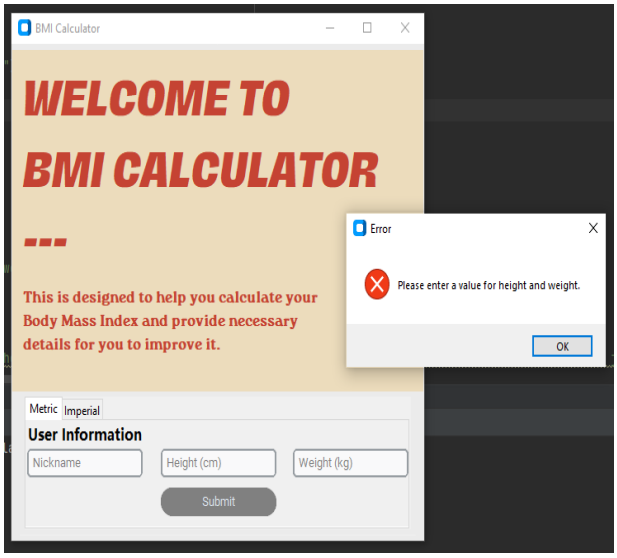
```
title_label = tkinter.Label(tab1, text="User Information", font=('Microsoft YaHei', 12, 'bold'))
title_label.grid(row=0, column=0)

nickname_entry = customtkinter.CTkEntry(master=tab1, placeholder_text="Nickname")
nickname_entry.grid(row=2, column=0)
height_entry = customtkinter.CTkEntry(master=tab1, placeholder_text="Height (cm)")
height_entry.grid(row=2, column=1)
weight_entry = customtkinter.CTkEntry(master=tab1, placeholder_text="Weight (kg)")
weight_entry.grid(row=2, column=2)

submit_button = customtkinter.CTkButton(master=tab1, text="Submit", border_width=0, corner_radius=15, fg_color="gray",
                                        command=calculate_bmi_m)
submit_button.grid(row=3, column=1, sticky="news", pady=10, padx=20)

feedback_label = tkinter.Label(top_frame, text="WELCOME TO\nBMI CALCULATOR\n---",
                               font=('Hagrid Text Trial Extrabold', 34, 'bold'),
                               bg="#ecdcbc", fg="#c54333", justify="left")
feedback_label.pack(padx=10, pady=(0, 0), anchor=W)

feedback_label = tkinter.Label(top_frame, text="This is designed to help you calculate your\nBody Mass Index and "
                                               "provide necessary\ndetails for you to improve it.\n",
                               font=('RocaOne-Rg', 12, 'bold'), bg="#ecdcbc", fg="#c54333", justify="left")
feedback_label.pack(padx=10, pady=(10, 10), anchor=W)

# User information in Imperial

title_label2 = tkinter.Label(tab2, text="User Information", font=('Microsoft YaHei', 12, 'bold'))
title_label2.pack(anchor=W)

entry_frame = tkinter.Frame(tab2)
entry_frame.pack()

nickname_entry2 = customtkinter.CTkEntry(master=entry_frame, placeholder_text="Nickname")
nickname_entry2.pack(side="left", padx=5)

height_entry2 = customtkinter.CTkEntry(master=entry_frame, placeholder_text="Feet", width=70)
height_entry2.pack(side="left", padx=5)

inches_entry = customtkinter.CTkEntry(master=entry_frame, placeholder_text="Inches", width=70)
inches_entry.pack(side="left", padx=5)

weight_entry2 = customtkinter.CTkEntry(master=entry_frame, placeholder_text="Weight (lbs)")
weight_entry2.pack(side="left", padx=5)

submit_button2 = customtkinter.CTkButton(master=tab2, text="Submit", border_width=0, corner_radius=15, fg_color="gray",
                                         command=calculate_bmi_i)
submit_button2.pack(pady=10)

# To add alignment to the widget
for widget in bottom_frame.winfo_children():
    widget.grid_configure(padx=10, pady=5)

window.mainloop()
```

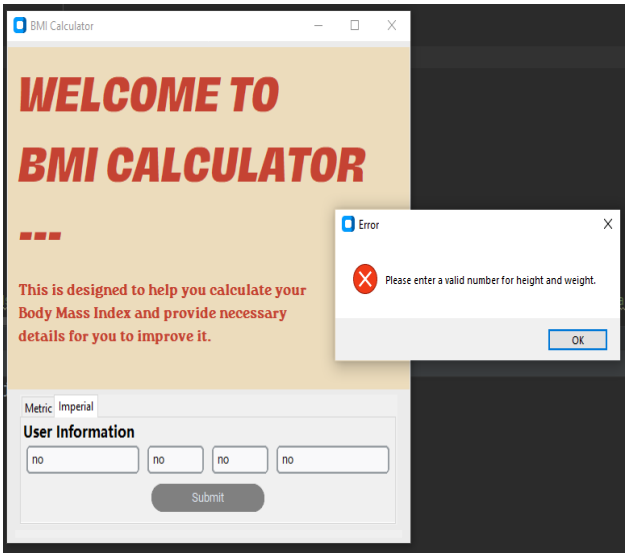
IV. PROGRAM OUTPUT (Frank)

Opening of the output:

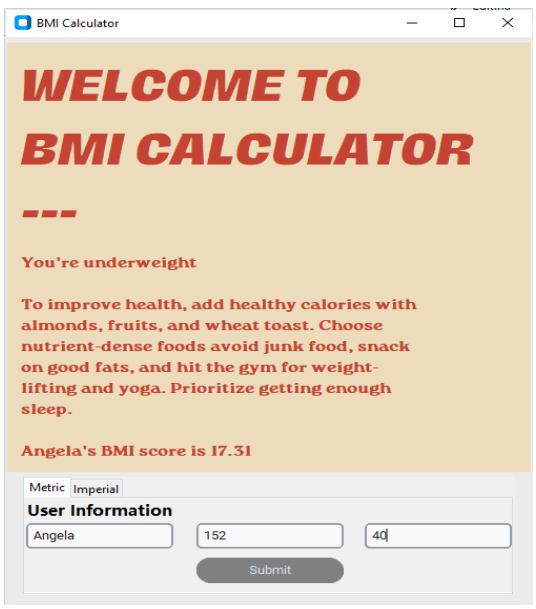


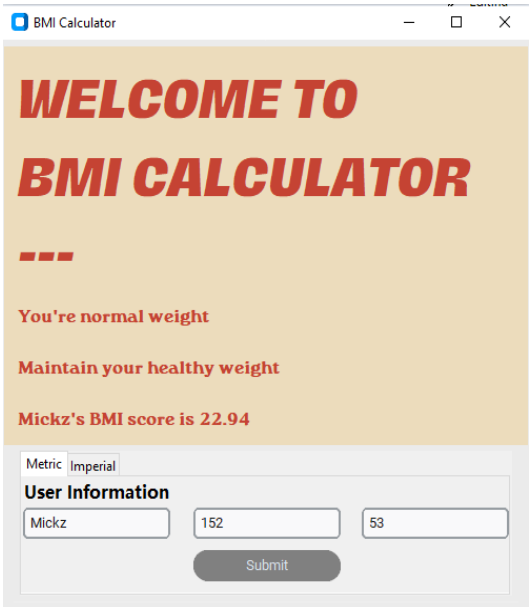
After running the code, we took at first glance to greet the user's presence and introducing the BMI Calculator. 2 measurement systems provide the opening of the output to calculate the user's Body Mass Index; the First tab is the Metric, and the Second tab is imperial. The first tab is provided an entry of the user's nickname, height, and weight or kilograms(kg), and the second tab is provided an entry of the user's nickname, feet and inches, and weight or pounds(lbs). Lastly, the GUI will provide a "submit" button that will calculate the user's inputted numbers of their Body Mass.

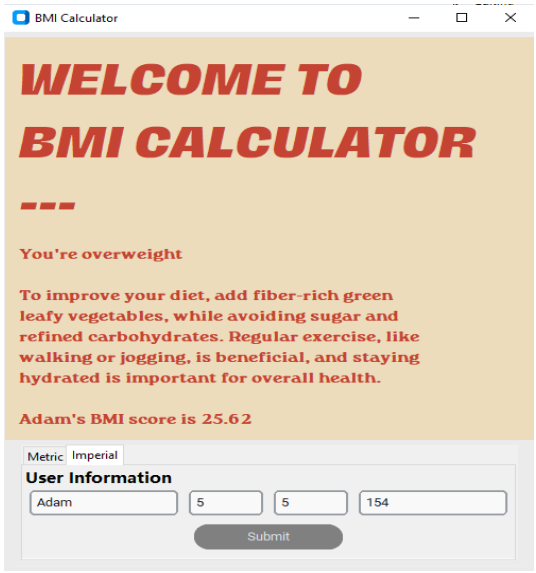




In this first output, the user will encounter an error message if the user did not input a valid requirement of these entries. The message box is only displayed if the user inputs a mistake or braindead. It works on both tabs (Metric, and Imperial) if the user inputs the wrong requirement.







There are 4 main outputs in our BMI Calculator:

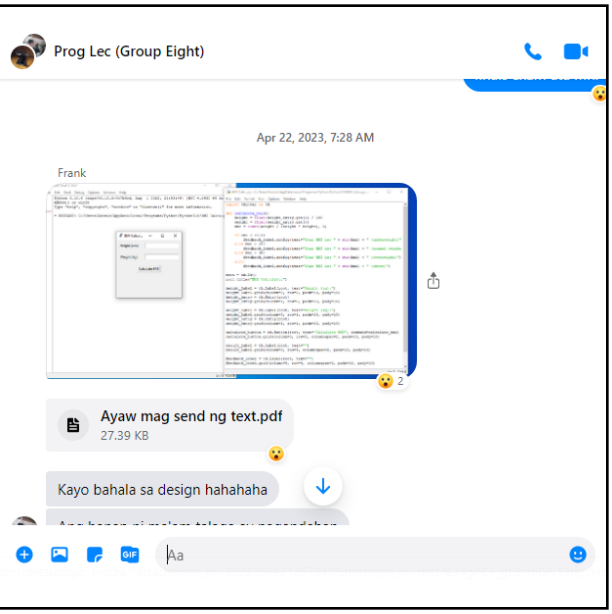
1. **Underweight Output** - If the user inputs a lower value of 18.5 his weight in metric or imperial, it will display a new text in the top frame which will provide some healthy tips for the user's underweight status.

2. **Normal Weight Output** - If the user inputs an average value or greater than 25 of his weight in metric or imperial, it will display the user's score and tell the user to keep up his healthy weight in the top of the frame.

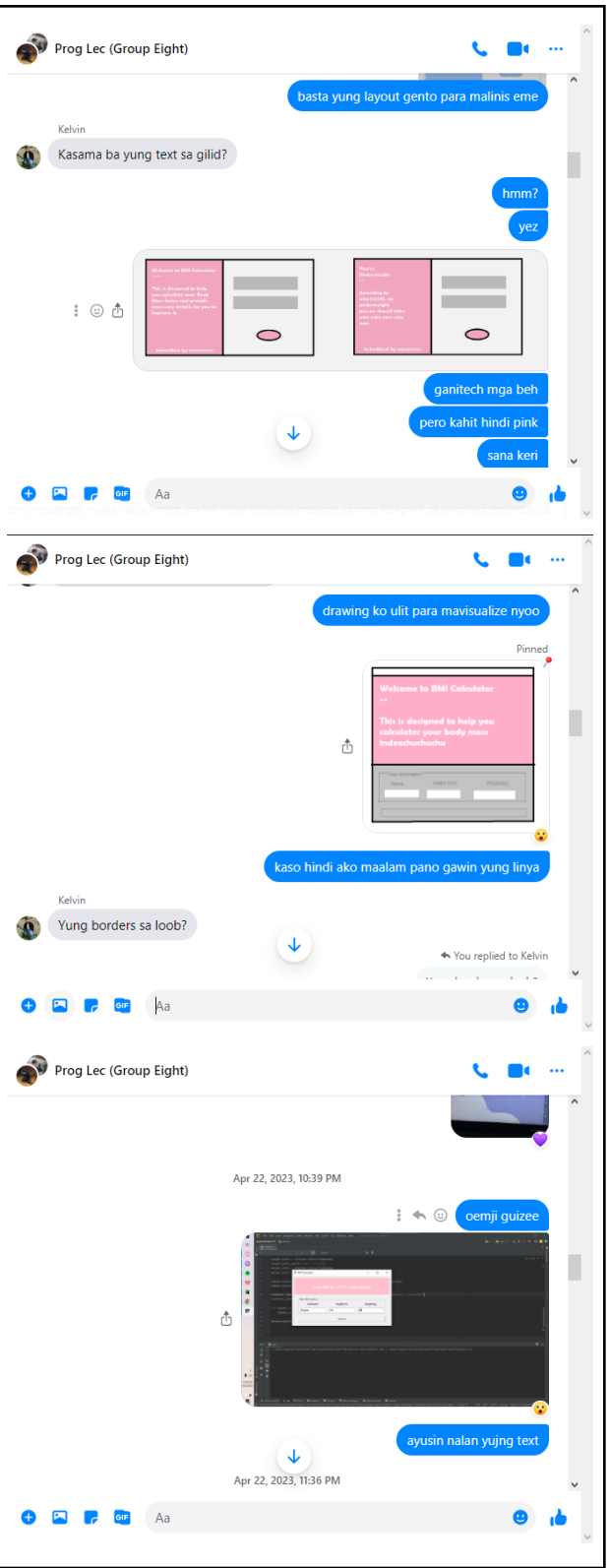
3. **Overweight Output** - If the user inputs a higher value or lower than 30 of his weight in metric or imperial, it will display the user's score and recommend a healthy tip of the user's weight at the top of the frame.

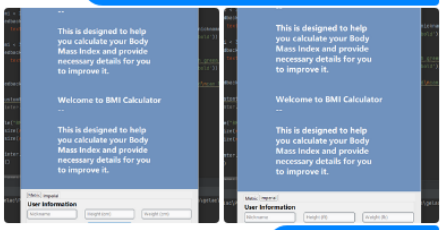
4. **Obese Output** - If the user inputs a value that is greater than 30 of his weight in metric or imperial, it will display the user's score and recommend a highly recommended healthy tip for obese users at the top of the frame.

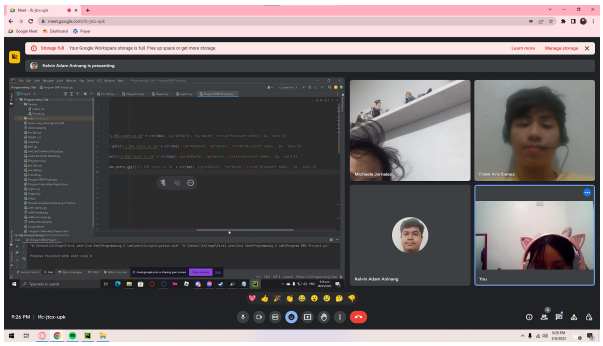
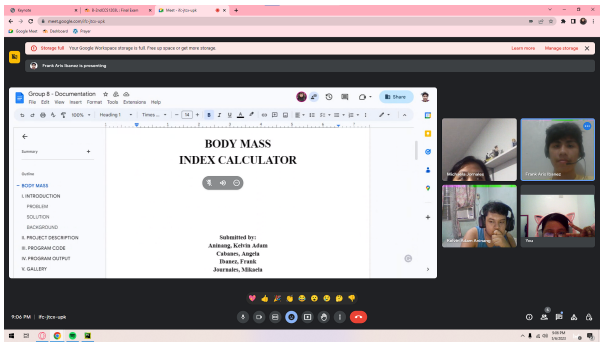
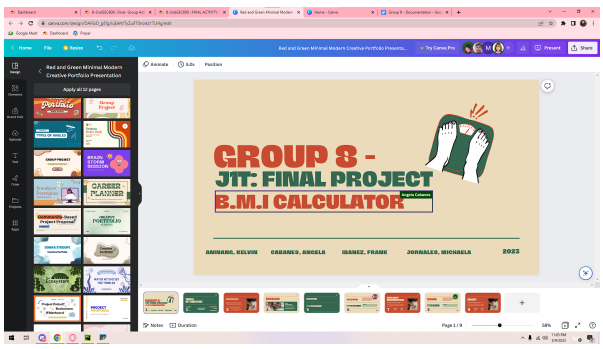
V. GALLERY

Frank Ibanez provided the base code	

Face to face meeting of the members	

Angela Cabanes provide the GUI draft design	

	Okie Okies na huehue sorry GUYS OEMJI MAY DINAGDAG AKO LUVETTE  cons: humaba code pros: mas functional sa user eme yungcode.txt 7.06 KB eto guys may dinilete ako real na real na real na talaga real final 
Michaela Jornales provided the contents inside the program.	Prog Lec (Group Eight) Mickz Okies Mickz Normal: "Maintain your healthy weight" Underweight: " 5 Helpful Tips to improve/ gain your weight: 1. Add healthy calories(healthy side dish almonds, fruits and wheat toast) 2. Go nutrient dense (reduced eating junk food) 3. Snacks!(snacks with good fats) 4. Hit the Gym(Try weightlifting and yoga) 5. Sleep the night away" Overweight: 5 Tips that can help you loss weight: 1. Add Fiber in your diet (green leafy vegetables) 2. Avoid Sugar 3. Avoid refined carbohydrates 4. Habitual Exercises (walking outside or jogging a little bit) 5. Stay Hydrated" Obese: "Obesity is unhealthy and can le[illegible]o heart stroke and other health complications, Seek doctor to know more about your current Aa

Kelvin Aninang debug every problem inside code encountered	
Members doing the document and the powerpoint presentation	 
Final Code/Design