BMI CALCULATOR

CAPSTONE PROJECT DOCUMENTATION

SUBMITTED BY

J.VANI SUMEDHA	23MSD7018
REIHA J R	23MSD7009
SNEHA M	23MSD7036
VAISHNAVI R	23MSD7028

in partial fulfilment for the award of the degree

of

Master of Science

in

Data Science

VIT-AP University , Andhra Pradesh

CONTRIBUTIONS BY THE TEAM

SNEHA.M	Front-end GUI Design and Slider Setup.
VAISHNAVI.R	Front-end Image Configuration and Additional UI elements
REIHA J.R	Back-end logic-BMI Calculation and BMI Category
J.VANI SUMEDHA	Report Display and Documentation.

.

PROBLEM STATEMENT

The BMI (Body Mass Index) app aims to provide a quick and easy way for individuals to calculate their BMI, a numerical measure of body fat based on height and weight.

The app addresses the challenge of promoting health awareness by providing a simple, accessible tool for individuals to assess their body mass status

In today's world, where lifestyle-related diseases are on the rise, monitoring one's health is crucial. BMI is a commonly used metric to assess whether an individual's weight is within a healthy range relative to their height.

Understanding BMI can contribute to the prevention of health issues associated with being underweight, overweight, or obese. The app fits into the broader context of encouraging individuals to be proactive about their health.

PROJECT DESCRIPTION

The BMI (Body Mass Index) Calculator app is developed to empower users with a simple yet effective tool for assessing their body mass status. The primary goal of the app is to promote health awareness by providing a user-friendly interface for BMI calculations and offering insights into the implications of weight relative to height.

GOALS AND OBJECTIVES:

HEALTH MONITORING

- Objective: Enable users to monitor their health by calculating BMI.
- Implementation: Users input their weight and height to obtain a quick BMI assessment.

HEALTH AWARENESS

- Objective: Increase awareness about the significance of BMI in relation to overall health.
- Implementation: The app categorizes BMI results and provides educational messages to inform users about their body mass status.

USER-FRIENDLY INTERFACE

- Objective: Create an intuitive and visually appealing interface for a positive user experience.
- Implementation: Incorporates graphical elements, sliders, and dynamic visual feedback to enhance user engagement.

INTERACTIVITY

- Objective: Allow users to interact with the app by adjusting height and weight slider.
- Implementation: Sliders dynamically update visual elements, providing an interactive and personalized experience.

MAIN FEATURES:

BMI CALCULATION:

- Users input weight and height, and the app calculates BMI using the standard formula.

VISUAL FEEDBACK:

- Categorizes BMI results (Underweight, Normal, Overweight, Obese) for easy interpretation.

- Provides informative messages to educate users about their body mass status.

USER INTERACTION:

- Entry boxes for manual input of height and weight.

- Sliders for interactive adjustments, offering a dynamic and engaging experience.

VISUAL ELEMENTS:

- Utilizes images, scales, and dynamically resized visuals to represent BMI results graphically.

- Enhances the visual appeal for a more attractive and user-friendly interface.

EDUCATIONAL COMPONENT:

- Incorporates educational messages to inform users about the importance of BMI in maintaining overall health.

SOURCE CODE:

```
from tkinter import *
import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk

root = Tk()
root.title("BMI Calculator")
root.geometry("470x580+300+200")
root.resizable(False, False)
root.configure(bg="#f0f1f5")

def BMI():
    h = float(Height.get())
    w = float(Weight.get())

    # convert height into meters
    m = h / 100
    bmi = w / m**2
    label1.config(text=f"{bmi:.2f}")
    if bmi <= 18.5:
        label2.config(text="Underweight!")
        label3.config(text="You have lower weight \nthan normal body!")
    elif 18.5 < bmi <= 25:
        label2.config(text="Normal!")
        label3.config(text="It indicates you are healthy!")
    elif 25 < bmi <= 30:
        label2.config(text="Overweight!")
        label3.config(text="It indicates that a person is \nslightly overweight.\nA doctor may advise losing \nsome weight for health reasons.")
    else:
        label2.config(text="Obese!")
        label3.config(text="Health may be at risk \nif they do not lose weight!")
# icon
image_icon = PhotoImage(file="C:/Users/reiha/Downloads/icon.png")
root.iconphoto(False, image_icon)
# top
top = PhotoImage(file="C:/Users/reiha/Downloads/top.png")
top_image = Label(root, image=top, background="#f0f1f5")
top_image.place(x=-10, y=-10)
# bottom box
Label(root, width=72, height=18, bg="Lightblue").pack(side=BOTTOM)


# two boxes
```

```
box = PhotoImage(file="C:/Users/reiha/Downloads/box.png")
Label(root, image=box).place(x=20, y=100)
Label(root, image=box).place(x=240, y=100)
# scale
scale = PhotoImage(file="C:/Users/reiha/Downloads/scale.png")
Label(root, image=scale, bg="lightblue").place(x=20, 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(get_current_value()))

    img = Image.open("C:/Users/reiha/Downloads/man.png")
    resized_image = img.resize((50, 10 + size))

    photo2 = ImageTk.PhotoImage(resized_image)
    secondimage.config(image=photo2)
    secondimage.place(x=70, y=550 - 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())



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 = StringVar()
Weight = StringVar()
height = Entry(root, textvariable=Height, width=5, font="arial 50", bg="#fff", fg="#000", bd=0, justify=CENTER)
height.place(x=35, y=160)
Height.set(get_current_value())

weight = Entry(root, textvariable=Weight, width=5, font="arial 50", bg="#fff", fg="#000", bd=0, justify=CENTER)
weight.place(x=255, y=160)
Weight.set(get_current_value2())
```

```
# man image
secondimage = Label(root, bg="lightblue")
secondimage.place(x=70, y=530)

Button(root, text="View Report", width=20, height=2, font="arial 10 bold", bg="#1f6e68", fg="white", command=BMI).place(x=300, y=65)

label1 = Label(root, font="arial 60 bold", bg="lightblue", fg="#fff")
label1.place(x=125, y=305)

label2 = Label(root, font="arial 20 bold", bg="lightblue", fg="#3b3a3a")
label2.place(x=280, y=430)

label3 = Label(root, font="arial 10 bold", bg="lightblue")
label3.place(x=200, y=500)

root.mainloop()
```

RESULTS AND DISCUSSION

- Successful BMI Calculation:

- The app successfully calculates BMI based on user input for weight and height.

- Immediate feedback is provided, categorizing BMI results and offering interpretations.

- User-Friendly Interface:

- The user interface is intuitive and visually appealing, incorporating graphical elements and dynamic visuals.

- Interactivity through sliders enhances user engagement and provides a personalized experience.

- Health Awareness Promotion:

- The app achieves its goal of promoting health awareness by including educational messages.

- Users receive insights into the implications of their body mass status and are encouraged to monitor their health proactively.

- Platform Compatibility:

- The app is developed with cross-platform compatibility, ensuring users on different systems can access and benefit from it.

- Technical Competence:

- Utilization of Tkinter and PIL libraries showcases technical competence in GUI development and image processing.

- Modular code structure facilitates future scalability and improvements.

CHALLENGES ENCOUNTERED

1. Image Processing Complexity:

- Integrating dynamically resized images based on BMI presented a challenge.

- Overcoming this required careful handling of image dimensions and ensuring responsiveness to user input.

2. Educational Content Balancing:

- Balancing informative messages without overwhelming users with excessive information.

SNAPSHOT

IMAGE 1 : CREATING THE SCALE FOR HEIGHT AND WEIGHT

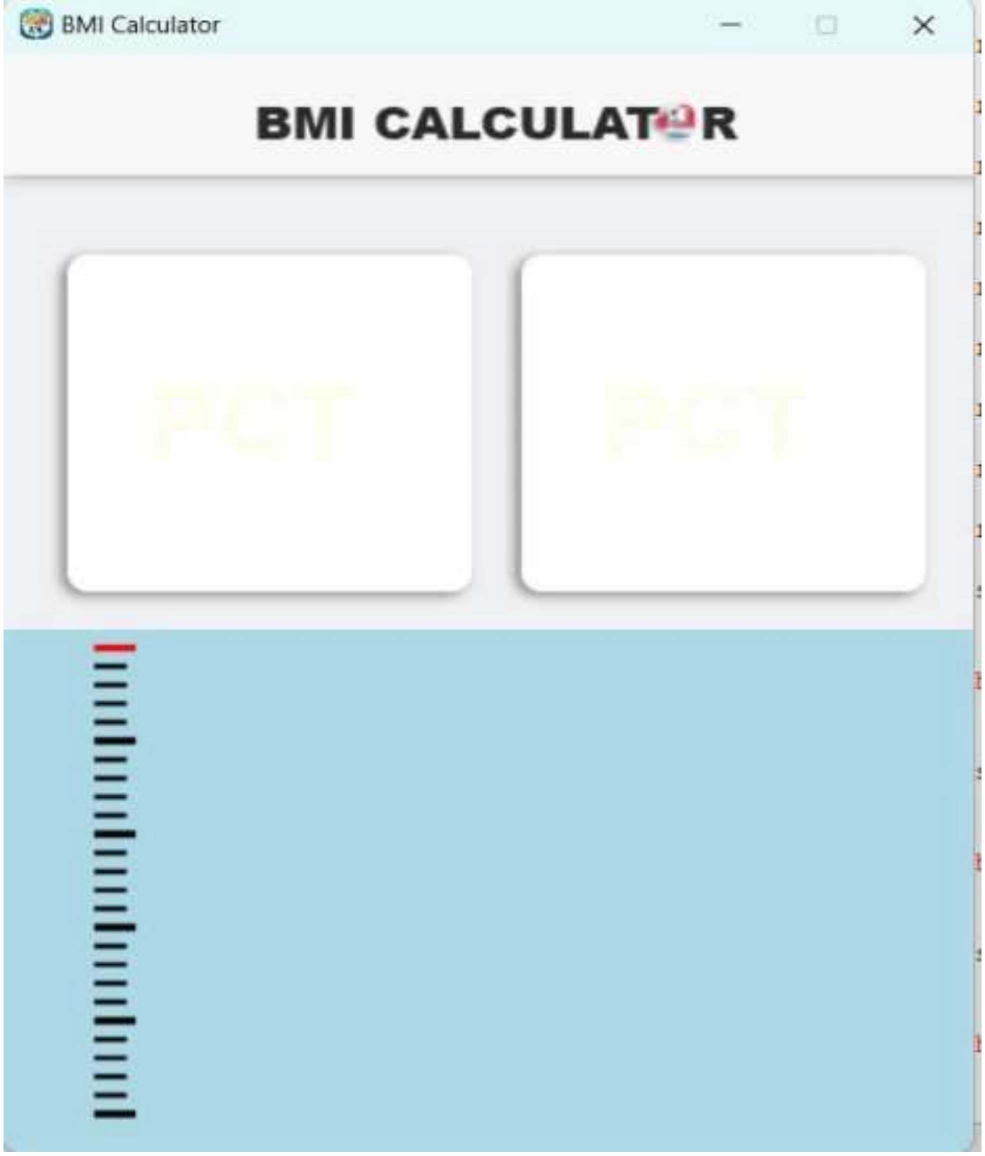


IMAGE -2: INITIALIZING THE VALUES FOR THE SCALE

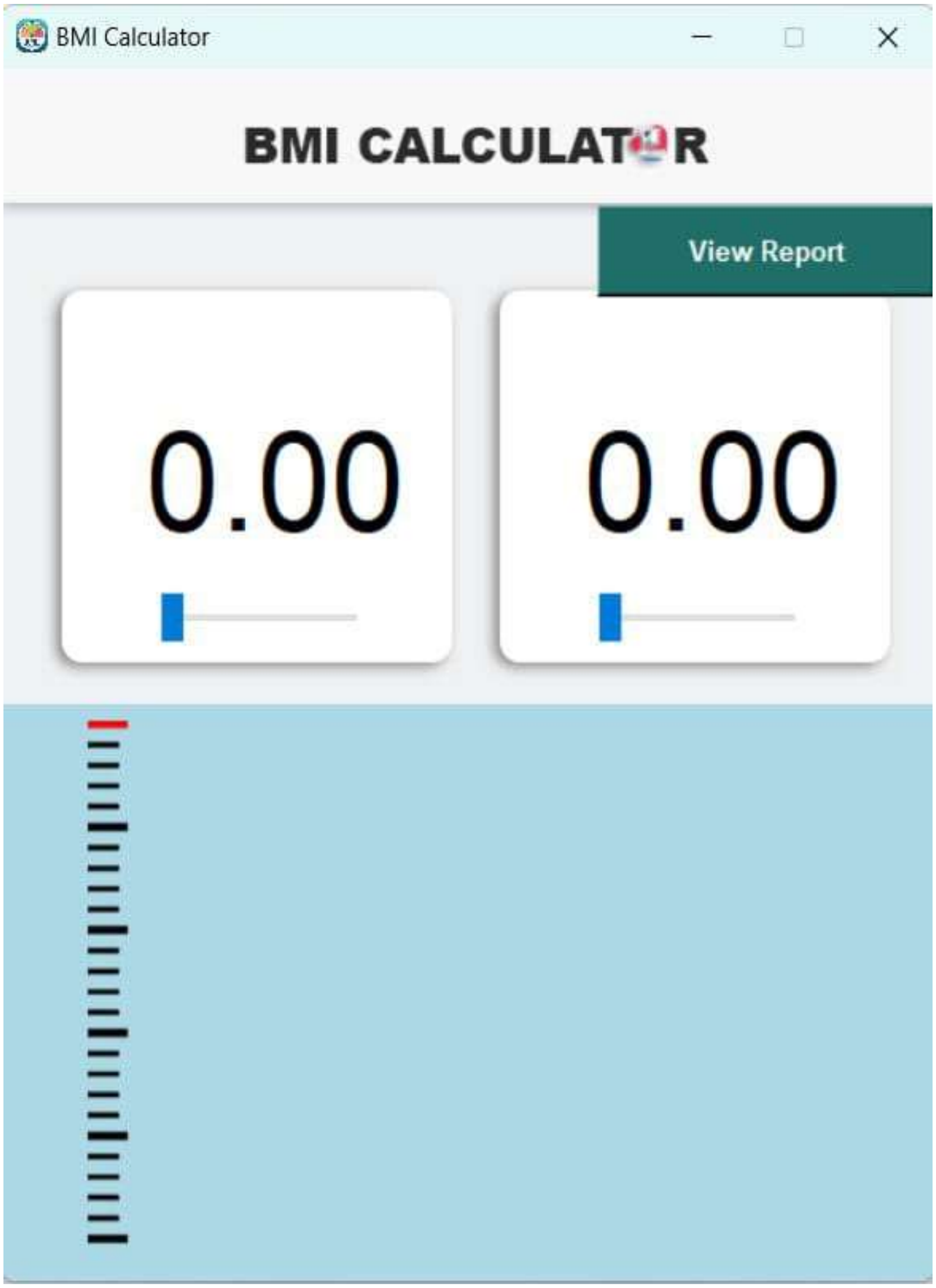
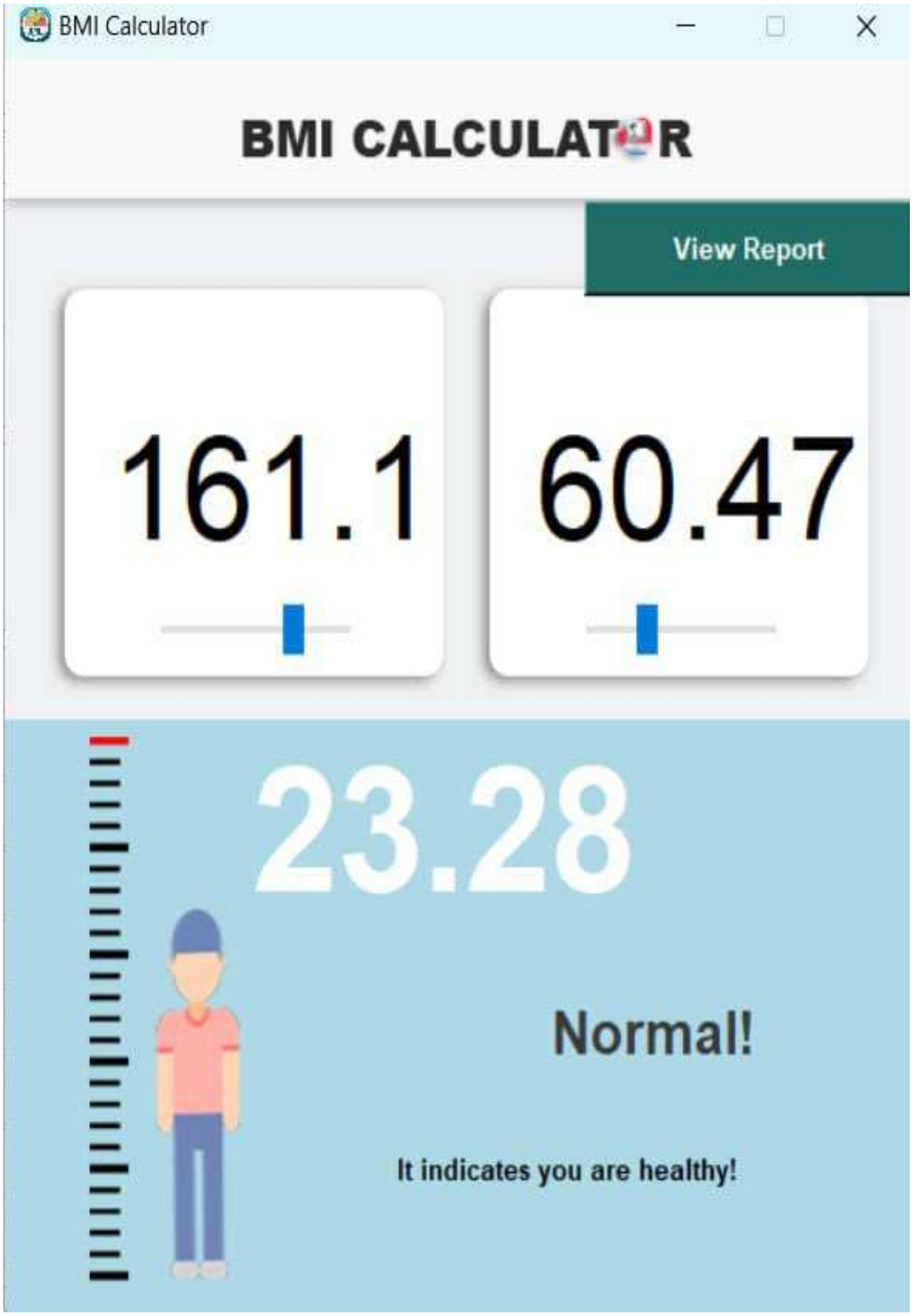


IMAGE-3 THE SCALE SHOWS THE CONDITION AS ‘NORMAL’FOR THE RESPECTIVE HEIGHT AND WEIGHT VALUE.



161.1

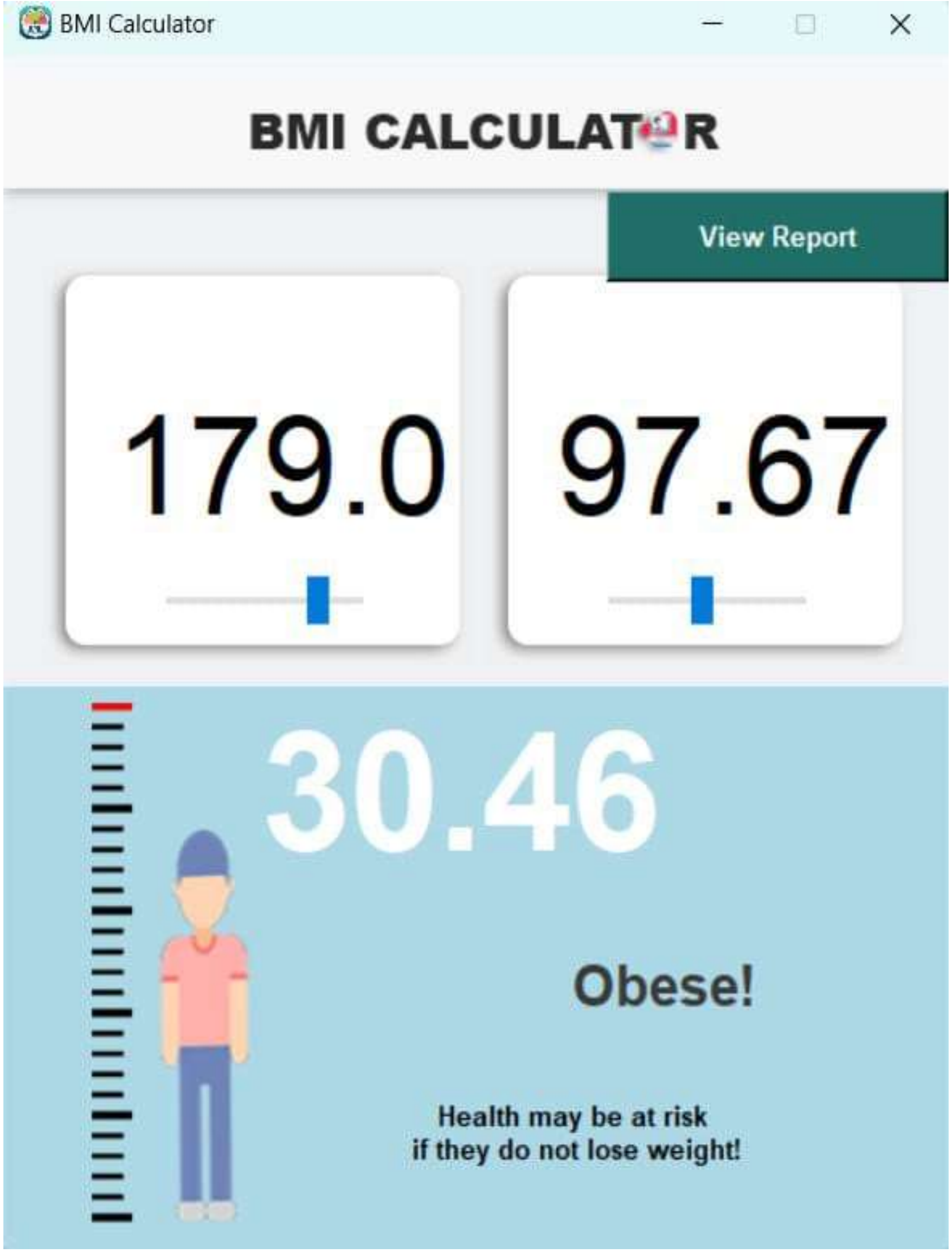
60.47

23.28

Normal!

It indicates you are healthy!

IMAGE-4 THE SCALE SHOWS THE CONDITION AS 'OBESE'FOR THE RESPECTIVE HEIGHT AND WEIGHT VALUE.



179.0

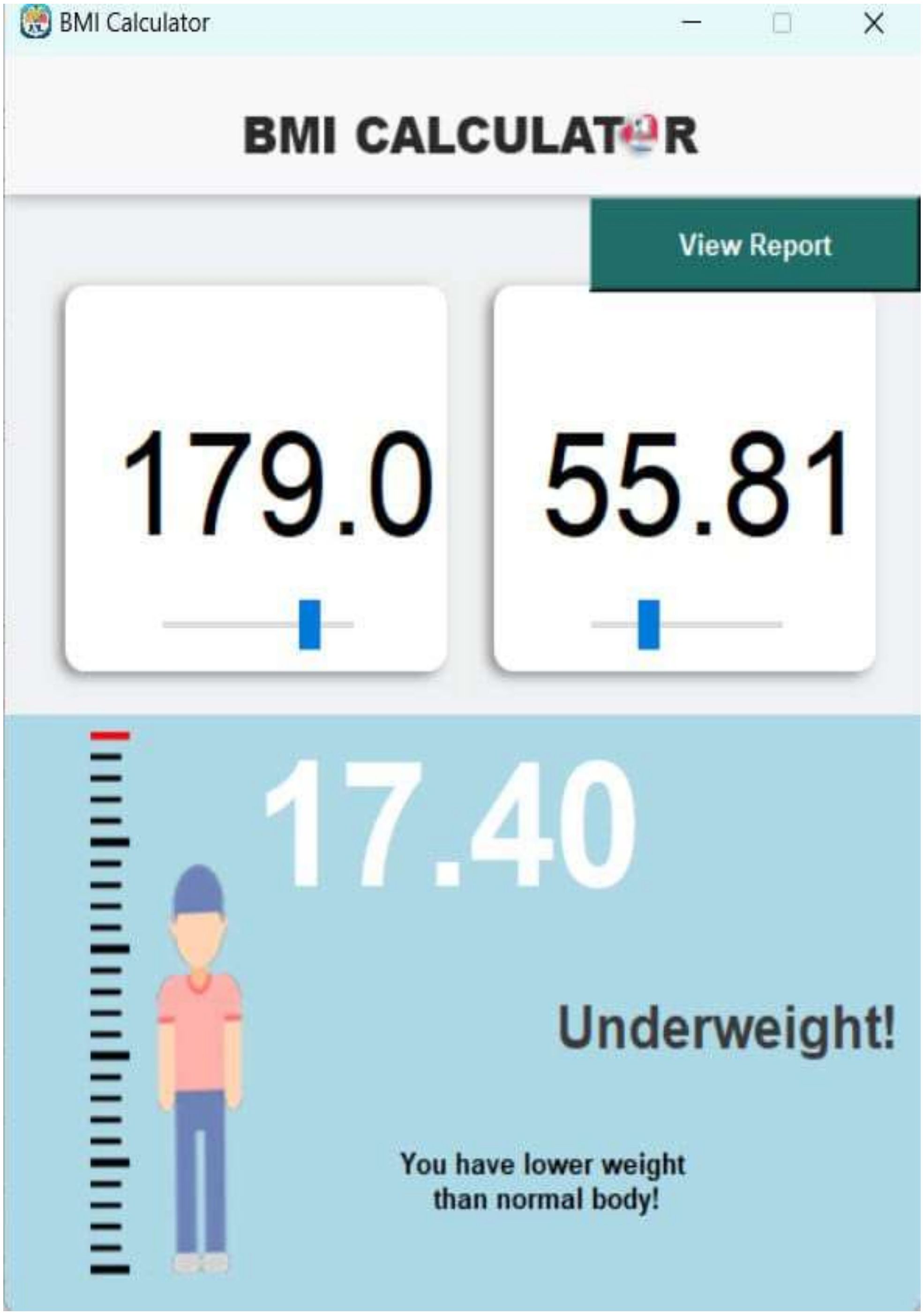
97.67

30.46

Obese!

Health may be at risk
if they do not lose weight!

IMAGE-5: THE SCALE SHOWS THE CONDITION AS ‘UNDERWEIGHT’FOR THE RESPECTIVE HEIGHT AND WEIGHT VALUE.



179.0

55.81

17.40

Underweight!

You have lower weight
than normal body!

CONCLUSION

The BMI Calculator app has achieved success in its key objectives, providing users with a user-friendly interface for accurate BMI calculations and promoting health awareness through informative messages. The app's positive outcomes include successful BMI calculation, an intuitive and engaging user interface, cross-platform compatibility, and the integration of educational content. However, challenges were encountered in balancing educational messages and dynamically resizing images based on BMI. Lessons learned emphasize the importance of user-centric design, iterative development, and thoughtful feature balancing. Overall, the project's impact is significant, offering individuals a practical tool for monitoring their health and empowering them with insights into their body mass status. The app's success and lessons learned contribute to its continuous improvement, aligning with user expectations and promoting a healthier lifestyle.

FUTURE IMPROVEMENTS:

- Integration of features to provide personalized health recommendations based on BMI results.
- Expansion of educational content to cover more health-related topics.
- Further refinement of visual elements for an engaging user experience.
- Addition of animations or transitions for a polished look.