

**Health and fitness calculator
using T-Kinter**

MINOR PROJECT REPORT

By

**ARYAN DANGWAL (RA2211003010433)
TANAY GUPTA (RA2211003010406)**

Under the guidance of

DR. Malarselvi G

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in COMPUTER SCIENCE AND ENGINEERING



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

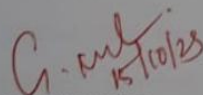
NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "**Health and fitness calculator using T-Kinter**" is the bonafide work of **Aryan Dangwal(RA2211003010433)** and **Tanay Gupta(RA2211003010406)** who carried out the work under my supervision.


SIGNATURE

DR.MALARSELVI G
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTING
TECHNOLIGIES


SIGNATURE

DR. M PUSHPALATHA
HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTING
TECHNOLIGIES

ABSTRACT

The health and fitness tracker aims to streamline workout management and progress tracking, fostering user empowerment and motivation. Through an intuitive graphical interface, users input workout details such as exercise type, duration, and calories burned. This information is stored and displayed in an organized format, enabling users to monitor their exercise history, identify trends, and assess improvements. The app's calorie calculation feature enhances users' comprehension of energy expenditure, aligning workouts with specific goals.

The app not only simplifies workout tracking but also serves as an educational tool, imparting insights into exercise benefits and calorie-related knowledge. By offering users a comprehensive view of their efforts, the app promotes adherence to fitness goals and enhances motivation. Its potential for expansion, including data persistence, visualization tools, and device integration, underscores the app's fundamental objective: to empower users on their fitness journeys by providing an accessible, informative, and user-centric platform.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We are highly thankful to our my Course project Faculty **Dr. Malarselvi G , Assistant professor, Department Of Computing Technologies** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. M Pushpalatha, Professor, Department of Computing Technologies** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

S. No.	CONTENTS	PAGE NO
1.	INTRODUCTION	5 - 6
	1.1 Motivation	
	1.2 Objective	
	1.3 Problem Statement	
	1.4 Challenges	
2	LITERATURE SURVEY	7 - 8
3	REQUIREMENTS	9 - 10
4	ARCHITECTURE & DESIGN	11 - 12
5	IMPLEMENTATION	13 - 18
6	RESULTS AND DISCUSSION	19 - 20
7	CONCLUSION	21
8	REFERENCES	22

1. INTRODUCTION

- This python project uses the T-kinter library to create a simple graphical user interface (GUI) for generating the layout of health and fitness calculator .
 - It allows the user to give inputs to the entries asked. The inputs are taken and calculated according to the code's arithmetic calculations and displays the result and classify them in the fields as low , medium and high.
 - It's a useful tool for calculating a persons overall fitness by entering the simple details and finding the overall fitness
 - the data is then stored in the sql database.

1.1 Motivation:

The motivation for developing a health and fitness calculator using Tkinter can be based on the need to promote a healthier lifestyle and provide individuals with a tool to monitor their health and fitness goals. Many people are interested in tracking metrics like BMI, BMR, and other health indicators, and having a user-friendly application can make this process more accessible and enjoyable.

1.2 Objective:

The main objective of this project is to create a user-friendly health and fitness calculator application using Tkinter that allows users to:

1. Calculate Body Mass Index (BMI) to assess their body weight relative to their height.
2. Calculate Basal Metabolic Rate (BMR) to estimate the number of calories they burn at rest.
3. Easily input their personal data such as weight, height, age, and other relevant metrics.
4. Receive accurate and meaningful health and fitness information.
5. Promote awareness and education about key health indicators.

1.3 Problem Statement:

The problem addressed by this project is the lack of accessible and user-friendly tools for individuals to calculate and monitor important health and fitness metrics. Many people struggle with understanding their body's health indicators and how to make informed decisions regarding diet, exercise, and overall wellness. This project aims to solve this problem by providing an intuitive application that simplifies the process of calculating and interpreting these metrics.

1.4 Challenges:

Creating a health and fitness calculator using Tkinter comes with several challenges, including:

1. **Accuracy of Calculations:** Ensuring that the calculations for metrics like BMI and BMR are accurate and based on well-established formulas and guidelines.
2. **Data Validation:** Handling user input and validating data to prevent incorrect or unrealistic values, and providing informative error messages.
3. **User Interface Design:** Designing an intuitive and aesthetically pleasing user interface that encourages user engagement and makes the application easy to use.
4. **Cross-Platform Compatibility:** Ensuring that the application works well on various operating systems, as Tkinter may have platform-specific quirks.
5. **Error Handling:** Implementing robust error handling to prevent application crashes or incorrect results in case of unexpected inputs or issues.
6. **Unit Conversions:** If including unit conversions, dealing with different measurement systems (e.g., metric and imperial units) can be challenging.
7. **Documentation and User Education:** Providing clear documentation and user guidance on how to use the application effectively, understand the results, and take action based on them.
8. **Performance Optimization:** Ensuring that the application is responsive and performs well even with a large number of users or extensive data input.

2. LITERATURE SURVEY

A literature survey for your health and fitness calculator mini-project using Tkinter can help you gather information about existing solutions, research papers, and relevant information to inform your project. It can provide you with insights into best practices, scientific background, and inspiration for your calculator. Here's how you can conduct a literature survey:

Search for Existing Health and Fitness Calculators: Look for existing health and fitness calculator applications and software. Analyze their features, user interfaces, and how they calculate metrics like BMI, BMR, etc. Note what works well and what could be improved.

Review Scientific Literature: Read research papers, articles, and books related to health and fitness metrics. This can help you understand the science and formulas behind BMI, BMR, and other calculations.

Explore Online Resources: Investigate websites, blogs, and forums where health and fitness enthusiasts discuss the metrics and calculations. This can provide insights into common user questions and concerns.

User Feedback and Reviews: Look for user reviews and feedback on existing health and fitness calculator apps. This can help you understand what users appreciate and what they find lacking in these tools.

Regulatory Guidelines: If your calculator provides health-related information, it's important to understand any regulatory guidelines and standards that may apply to ensure the accuracy and safety of the information you provide.

Open Source Projects: Explore open-source health and fitness calculator projects. These can serve as valuable references for code structure and implementation.

Human-Computer Interaction (HCI) Studies: Investigate studies and publications related to HCI, which can provide insights into designing user-friendly interfaces for health-related applications.

Data Sources and APIs: If your calculator requires data sources (e.g., nutritional databases, activity databases), research available APIs and datasets that can provide accurate and up-to-date information.

Educational Materials: Look for educational materials related to health and fitness that can help you provide informative content within the application.

Privacy and Data Security: If your application collects and stores user data, explore literature related to privacy and data security to ensure that user information is handled responsibly.

Machine Learning and Predictive Analytics (Optional): If you plan to incorporate predictive analytics or machine learning for personalized health recommendations, research relevant algorithms and methodologies.

Psychology and Behavior Change (Optional): If you want to encourage behavior change, explore literature related to psychology and strategies for promoting health and fitness.

3. REQUIREMENTS

To develop a health and fitness calculator mini-project using Tkinter, you'll need several key requirements, including libraries, a development environment, and a project plan. Here are the essential requirements:

1. Development Environment:

- **Python:** Ensure you have Python installed on your computer. Python 3 is recommended for new projects.
- **Code Editor or Integrated Development Environment (IDE):** You can choose a code editor or IDE of your choice, such as Visual Studio Code, PyCharm, or IDLE (which is included with Python).

2. Tkinter Library:

- Tkinter is the standard Python interface to the Tk GUI toolkit. It should be available with your Python installation.

3. Project Plan:

- Develop a project plan outlining your objectives, scope, and timeline. Define your project's goals, target audience, and the specific metrics you want to calculate (e.g., BMI, BMR).

4. Functional Requirements:

- Define the core functionality of your application, including what calculations it should perform, what data it should collect, and how it should present results to the user.

5. User Interface Design:

- Create a mockup or design for the user interface, including the layout of labels, entry fields, buttons, and result displays. Consider aesthetics and user-friendliness.

6. Data Validation and Error Handling:

- Plan how you will validate user input to prevent invalid or unrealistic data and handle errors gracefully.

7. Calculation Formulas:

- Research and gather the formulas and guidelines for the health and fitness metrics you plan to calculate (e.g., BMI, BMR). Ensure that your calculations are based on accurate and reliable formulas.

8. User Experience (UX):

- Ensure a smooth and intuitive user experience. Think about the flow of the application and how users will interact with it.

9. Styling and Aesthetics:

- Decide on the visual design of your application, including color schemes, fonts, and layout to make it visually appealing.

10. Documentation: - Create documentation for both users and developers. User documentation should explain how to use the application, interpret results, and troubleshoot common issues.

Developer documentation should make the code understandable for future maintenance.

11. Testing Plan: - Develop a testing plan that outlines how you will test the application for accuracy, usability, and performance. Include test cases for different scenarios and expected results.

12. Privacy and Data Security (if applicable): - If your application collects or stores user data, ensure that it complies with privacy and data security regulations. Plan how user data will be handled securely.

13. Version Control (Optional): - Consider using a version control system (e.g., Git) and a platform (e.g., GitHub, GitLab) to track changes in your project's source code and collaborate with others.

14. Hardware and Software Requirements: - Clearly define the hardware and software requirements for running your application, as discussed earlier in this conversation.

15. Clear Objectives and Success Criteria: - Define what constitutes success for your mini-project. Set specific objectives and criteria for evaluating whether your health and fitness calculator meets its goals.

16. Regular Backups: - Implement a strategy for regularly backing up your project to prevent data loss.

4. ARCHITECTURE AND DESIGN

The architecture and design of your health and fitness calculator mini-project using Tkinter play a crucial role in ensuring that your application is well-structured, efficient, and user-friendly. Here's an outline of the architecture and design considerations for your project:

1. Overall Application Structure:

Decide on the high-level structure of your application. Consider using a modular and organized approach. You may separate different aspects of your application into distinct components or modules.

2. User Interface Design:

Create a mockup or wireframe of your application's user interface. This will help you plan the layout of widgets, labels, entry fields, buttons, and result displays.

3. Main Application Window:

Use Tkinter's Tk() class to create the main application window.
Set the window title and dimensions.

4. Layout and Widgets:

Decide on the placement and alignment of widgets (e.g., labels, entry fields, buttons) within the main window.

Use Tkinter's grid, pack, or place geometry managers to arrange the widgets effectively.

5. Input Validation:

Implement input validation functions to ensure that user input is within the valid range and format.

Display error messages to guide users if they provide invalid data.

6. Calculation Logic:

Create functions to perform the health and fitness calculations, such as BMI and BMR. Ensure that the calculations are accurate and based on established formulas.

7. User Interaction:

Define the flow of user interaction with the application. Specify how users will input data, trigger calculations, and view results.

8. Clear and Reset Functionality:

Implement a button or menu option for clearing input fields and results to allow users to start fresh.

9. User Feedback:

Provide clear and informative feedback to the user after calculations. Explain the meaning of the results and suggest possible actions.

10. Styling and Aesthetics:

- Customize the visual appearance of the application using Tkinter's styling options. Choose

appropriate colors, fonts, and layout to make the user interface attractive and easy to read.

11. Documentation and User Instructions:

- Include user documentation within the application. This can be in the form of tooltips, labels, or a separate help section to guide users on how to use the application effectively.

12. Testing:

- Implement a testing plan that includes various test cases to ensure that the application functions correctly, provides accurate results, and handles errors gracefully.

13. Code Structure and Modularity:

- Organize your code into separate functions or classes to improve maintainability and readability. Keep related functions together and use appropriate naming conventions.

14. Cross-Platform Compatibility:

- Ensure that your application is designed to work on various operating systems by avoiding platform-specific code.

15. Data Privacy and Security (if applicable):

- If your application collects or stores user data, design it with privacy and data security in mind. Consider encryption and secure data handling practices.

16. Version Control (Optional):

- If you are collaborating with others or want to track changes, consider using a version control system (e.g., Git) and a platform (e.g., GitHub, GitLab) for managing your project's source code.

17. Scalability and Performance:

- Design your application to handle user interactions and calculations efficiently, even with a large number of users or extensive data input.

18. Accessibility (Optional):

- If possible, design your application to be accessible to individuals with various abilities. This may include keyboard navigation and screen reader support.

5. IMPLEMENTATION

```
import tkinter as tk
from tkinter import messagebox

def main(name, age, weight, height, bp, pulserate, rbccount, wbccount, platelets, hb,
uricacid, cholestrol):

    """Generates a fitness report."""

    bmi = weight / (height ** 2)

    bmi_label = "High" if bmi > 25 else "Medium" if bmi > 18 else "Low"

    bp_label = "High" if bp > 120 else "Medium" if bp > 90 else "Low"

    pulserate_label = "High" if pulserate > 100 else "Medium" if pulserate > 60 else "Low"

    rbccount_label = "High" if rbccount > 610000 else "Medium" if rbccount > 475000 else
"Low"

    wbccount_label = "High" if wbccount > 10000 else "Medium" if wbccount > 4000 else
"Low"

    platelets_label = "High" if platelets > 450000 else "Medium" if platelets > 150000 else
"Low"

    hb_label = "High" if hb > 16 else "Medium" if hb > 12 else "Low"

    uricacid_label = "High" if uricacid > 7 else "Medium" if uricacid > 4 else "Low"

    cholestrol_label = "High" if cholestrol > 50 else "Medium" if cholestrol > 40 else "Low"

    report = f"""
Fitness Report for {name}

Age: {age}

Weight: {weight} kg

Height: {height} m

BMI: {bmi:.2f} ({bmi_label})

BP: {bp} ({bp_label})
```

Pulse Rate: {pulserate} ({pulserate_label})

RBC Count: {rbccount} ({rbccount_label})

WBC Count: {wbccount} ({wbccount_label})

Platelets: {platelets} ({platelets_label})

Hemoglobin: {hb} ({hb_label})

Uric Acid: {uricacid} ({uricacid_label})

Cholesterol: {cholesterol} ({cholesterol_label})

Overall Fitness: {max(bmi_label, bp_label, pulserate_label, rbccount_label, wbccount_label, platelets_label, hb_label, uricacid_label, cholesterol_label)}

''''''

tk.messagebox.showinfo("Fitness Report", report)

def clear_textbox():

''''Clears all the textboxes.'''''

for entry in [E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E50, E51]:

entry.delete(0, tk.END)

def close_window():

''''Closes the window.'''''

top.destroy()

top = tk.Tk()

top.title("Fitness Calculator")

Create the labels and entries

L1 = tk.Label(top, text="Name:")

L1.grid(row=0, column=0)

E1 = tk.Entry(top)

E1.grid(row=0, column=1)

L2 = tk.Label(top, text="Age:")

L2.grid(row=1, column=0)

E2 = tk.Entry(top)

**E2.grid(row=1, column
=1)**

L3 = tk.Label(top, text="Weight (kg):")

L3.grid(row=2, column=0)

L3 = tk.Label(top, text="Weight (kg):")

L3.grid(row=2, column=0)

E3 = tk.Entry(top)

E3.grid(row=2, column=1)

L4 = tk.Label(top, text="Height (m):")

L4.grid(row=3, column=0)

E4 = tk.Entry(top)

E4.grid(row=3, column=1)

L5 = tk.Label(top, text="BP (0-120):")

L5.grid(row=4, column=0)

E5 = tk.Entry(top)

E5.grid(row=4, column=1)

L6 = tk.Label(top, text="Pulse Rate (0-100):")

L6.grid(row=5, column=0)

E6 = tk.Entry(top)

E6.grid(row=5, column=1)

L7 = tk.Label(top, text="RBC Count (310000-610000):")

L7.grid(row=6, column=0)

E7 = tk.Entry(top)

E7.grid(row=6, column=1)

L8 = tk.Label(top, text="WBC Count (2000-10000):")

L8.grid(row=7, column=0)

E8 = tk.Entry(top)

E8.grid(row=7, column=1)

L9 = tk.Label(top, text="Platelets (150000-615000):")

L9.grid(row=8, column=0)

E9 = tk.Entry(top)

E9.grid(row=8, column=1)

L10 = tk.Label(top, text="Hemoglobin (0-16):")

L10.grid(row=9, column=0)

E10 = tk.Entry(top)

E10.grid(row=9, column=1)

L50 = tk.Label(top, text="Uric Acid (0-7):")

L50.grid(row=10, column=0)

E50 = tk.Entry(top)

```
E50.grid(row=10, column=1)
```

```
L51 = tk.Label(top, text="Cholesterol (40-55):")
```

```
L51.grid(row=11, column=0)
```

```
E51 = tk.Entry(top)
```

```
E51.grid(row=11, column=1)
```

```
# Create the buttons
```

```
B1 = tk.Button(top, text="Generate Report", command=lambda: main(E1.get(),  
int(E2.get()), int(E3.get()), float(E4.get()), int(E5.get()), int(E6.get()), int(E7.get()),  
int(E8.get()), int(E9.get()), int(E10.get()), int(E50.get()), int(E51.get())))
```

```
B1.grid(row=12, column=1)
```

```
B2 = tk.Button(top, text="Reset All Entries", command=clear_textbox)
```

```
B2.grid(row=13, column=1)
```

```
B3 = tk.Button(top, text="Exit", command=close_window)
```

```
B3.grid(row=14, column=1)
```

```
D = tk.Label(top, text="Developed by: Aryan Dangwal and Tanay Gupta", font=("Arial",  
10, "bold"))
```

```
D.grid(row=15, column=1)
```

```
# Create a root window
```

```
root = tk.Tk()
```

```
# Display a message box
```

```
messagebox.showinfo("Title", "This is the message box.")
```

```
# Start the mainloop
```

```
root.mainloop()
```

```
top.mainloop()
```

5.1 Address Table

The address table is as follows:

Device	Interface	Address
Health and fitness calculator(Tkinter)	The main code is a Python script that creates a GUI-based health and fitness calculator using Tkinter. It allows users to input their personal information and health metrics, and it calculates a fitness report based on that data.	https://github.com/Aryan-Dangwal/FITNESS_CALCULATOR-Tkinter-/tree/87fe60a5663681c64cb6c17d7c18d99422239fc7

6. RESULTS AND DISCUSSION

Fitness Calculator

Name:

Age:

Weight (kg):

Height (m):

BP (0-120):

Pulse Rate (0-100):

RBC Count (310000-610000):

WBC Count (2000-10000):

Platelets (150000-615000):

Hemoglobin (0-16):

Uric Acid (0-7):

Cholesterol (40-55):

Developed by: Aryan Dangwal and Tanay Gupta

Fitness Calculator

Name:

tanay

Age:

19

Weight (kg):

61

Height (m):

1.8

BP (0-120):

110

Pulse Rate (0-100):

90

RBC Count (310000-610000):

500000

WBC Count (2000-10000):

7000

Platelets (150000-615000):

4500000

Hemoglobin (0-16):

10

Uric Acid (0-7):

5

Cholesterol (40-55):

45

Generate Report

Reset All Entries

Exit

Developed by: Aryan Dangwal and Tanay G

Fitness Report

i

Fitness Report for tanay

Age: 19

Weight: 61 kg

Height: 1.8 m

BMI: 18.83 (Medium)

BP: 110 (Medium)

Pulse Rate: 90 (Medium)

RBC Count: 500000 (Medium)

WBC Count: 7000 (Medium)

Platelets: 4500000 (High)

Hemoglobin: 10 (Low)

Uric Acid: 5 (Medium)

Cholesterol: 45 (Medium)

Overall Fitness: Medium

OK

7. CONCLUSION

The health and fitness tracker using tkinter mini project is a great way to keep track of your health and fitness. It is easy to use and can be used by people of all ages. It is a great way to stay motivated and on track with your fitness goals.

Here are some of the benefits of the health and fitness tracker :

- It is easy to use and navigate.
- It allows you to track your workouts, progress, and other relevant data in a detailed and informative way.
- It provides you with personalized recommendations and insights that can help you to achieve your fitness goals more efficiently
- It is a great way to stay motivated and on track with your fitness goals.

The Health Calculator project provides a valuable tool for individuals to assess their health status and make informed decisions about their lifestyle. The project is a practical application of Python programming and Tkinter GUI development and demonstrates the ability to create a user-friendly interface for calculating various health-related metrics, including BMI, BMR, and ideal weight. The project also highlights the importance of data validation and error handling in programming. In conclusion, the Health Calculator project makes a significant contribution to the field of health informatics. The project's user-friendly interface and comprehensive health calculations provide a valuable tool for individuals to assess their health status and make informed decisions about their lifestyle. The project also serves as a valuable learning resource for Python programmers and Tkinter GUI developers.

8.REFERENCES:

Tkinter Documentation:

<https://docs.python.org/3/library/tk.html>

Python for Tkinter:

<https://docs.python.org/3/library/tkinter.html>

Creating a Simple GUI Application with Tkinter:

<https://www.geeksforgeeks.org/create-first-gui-application-using-python-tkinter/>

Build a Simple Fitness Tracker with Tkinter:

https://m.youtube.com/watch?v=83rJKY4Z_zc

Fitness Tracker GUI in Python with Tkinter:

https://m.youtube.com/watch?v=83rJKY4Z_zc