Title

"Fitness Tracker: A Comprehensive Health Monitoring Solution"

Introduction

Fitness trackers are widely used to track daily fitness data, such as steps taken, calories burned, heart rate, and sleep patterns. In this hackathon challenge, we have developed a fitness tracker using MATLAB and MATLAB Mobile. Our solution integrates multiple features to provide users with a holistic view of their health and fitness levels.

Key Features

1. Heart Rate Monitor
    - Description : Real-time heart rate monitoring using simulated ECG-like signals.
    - Functionality :
        - Detects full waveforms in the signal to count heartbeats.
        - Calculates heart rate in beats per minute (BPM).
        - Displays health status based on heart rate ranges:
            - 45–65 BPM : Healthy
            - 35–55 BPM : Normal
            - 25–35 BPM : Not Healthy
        - Provides visual feedback through a dynamic graph.
2. Accelerometer Integration
    - Description : Measures physical activity using accelerometer data.
    - Functionality :
        - Captures acceleration data from the device's built-in accelerometer.
        - Analyzes movement patterns to estimate steps taken or activity levels.
        - Displays activity metrics in real time.
3. Body Mass Index (BMI) Checker
    - Description : Calculates BMI based on user input (height and weight).
    - Functionality :
        - Allows users to input their height and weight.
        - Computes BMI using the formula: BMI=Height (m)2Weight (kg) .
        - Classifies BMI into categories:
            - Underweight: < 18.5
            - Normal: 18.5–24.9
            - Overweight: 25–29.9

        - Obese: ≥ 30
    - Provides personalized health recommendations based on BMI.

Technical Implementation

1. Heart Rate Monitor

- Signal Simulation : Simulated ECG-like signals are generated using a sine wave function.
- Peak Detection : Identifies peaks in the signal to count heartbeats.
- Heart Rate Calculation : Uses the time difference between consecutive peaks to calculate BPM.
- Health Status Classification : Based on predefined ranges, the system categorizes heart rate as "Healthy," "Normal," or "Not Healthy."

2. Accelerometer Integration

- Data Collection : Utilizes MATLAB Mobile's **mobiledev** object to access the device's accelerometer.
- Activity Analysis : Processes accelerometer data to detect motion patterns and estimate steps or activity levels.
- Real-Time Feedback : Displays activity metrics dynamically.

3. BMI Checker

- User Input : Collects height and weight inputs via UI components.
- Calculation : Implements the BMI formula to compute the result.
- Classification : Categorizes BMI into standard health ranges.
- Recommendations : Provides personalized advice based on BMI classification.

UI Design

The application is designed using MATLAB App Designer, featuring the following components:

- Main Dashboard : Displays real-time heart rate, activity level, and BMI.
- Heart Rate Monitor : Shows a dynamic graph of the ECG-like signal with peak detection.
- Accelerometer Section : Visualizes accelerometer data and activity metrics.
- BMI Calculator : Includes input fields for height and weight, along with a results section.

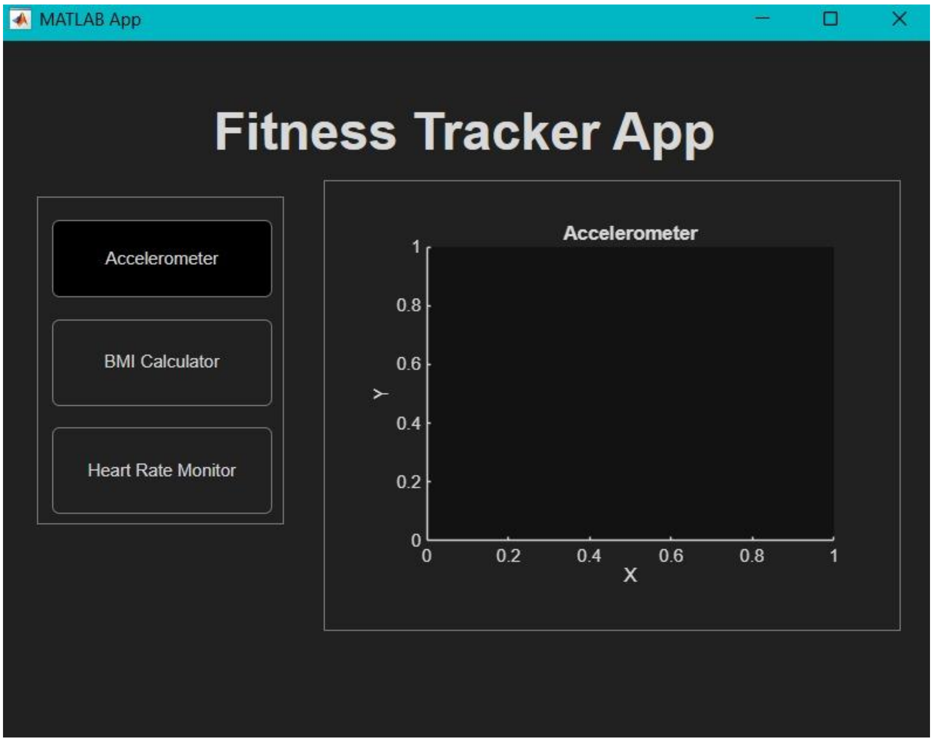
Innovative Aspects

1. Real-Time Data Processing : All features process data in real time, providing immediate feedback to the user.
2. Interdisciplinary Approach : Combines signal processing, sensor integration, and user interface design.

3. Personalized Health Insights : Offers tailored recommendations based on heart rate, activity, and BMI.

Future Enhancements

1. Integration with Wearables : Extend the app to work with wearable devices for more accurate data collection.
2. Machine Learning : Implement machine learning models to predict health trends based on historical data.
3. Cloud Connectivity : Store and analyze data in the cloud for long-term health tracking.

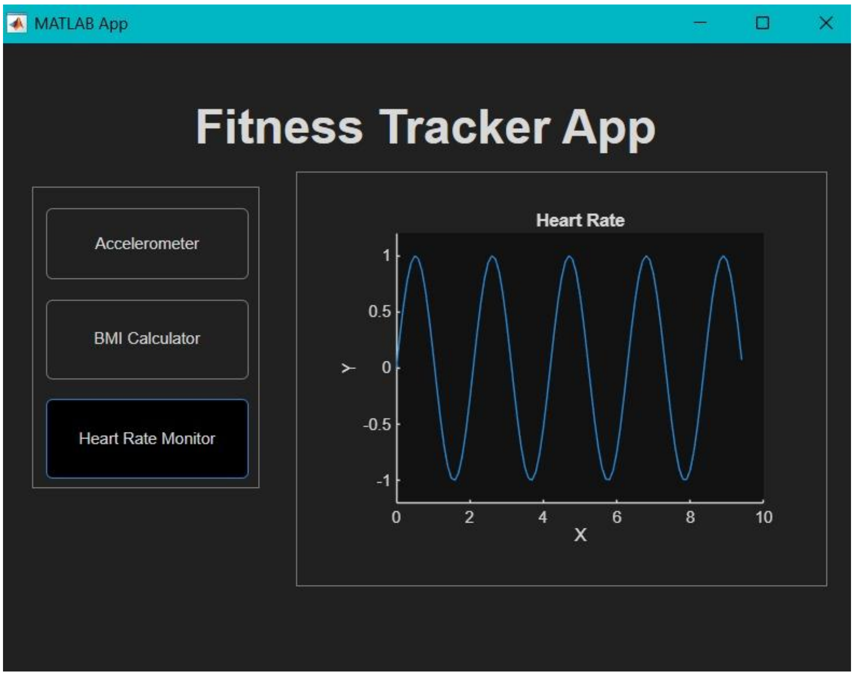


When the "Accelerometer" button is selected (as indicated by its darker background), the main content area on the right transforms to show an empty plot.

Functionality: This area is designed to visualize real-time or recorded accelerometer data. Once connected to an accelerometer sensor (e.g., from a smartphone or wearable device), the app would plot the acceleration data along the X, Y, and Z axes (though only a generic plot is shown here without specific axes labels beyond 'X' and 'Y' for demonstration purposes).

Purpose: This feature allows users to track movement, activity levels, or even specific motions by analyzing the acceleration patterns. It's a crucial component for activities like step counting, sleep tracking, or analyzing exercise intensity.

Currently, the plot is empty, indicating that no accelerometer data is being streamed or staying still.



This app is designed to monitor and visualize key health and fitness parameters using MATLAB App Designer. It features a clean, user-friendly interface with three main components:

Accelerometer

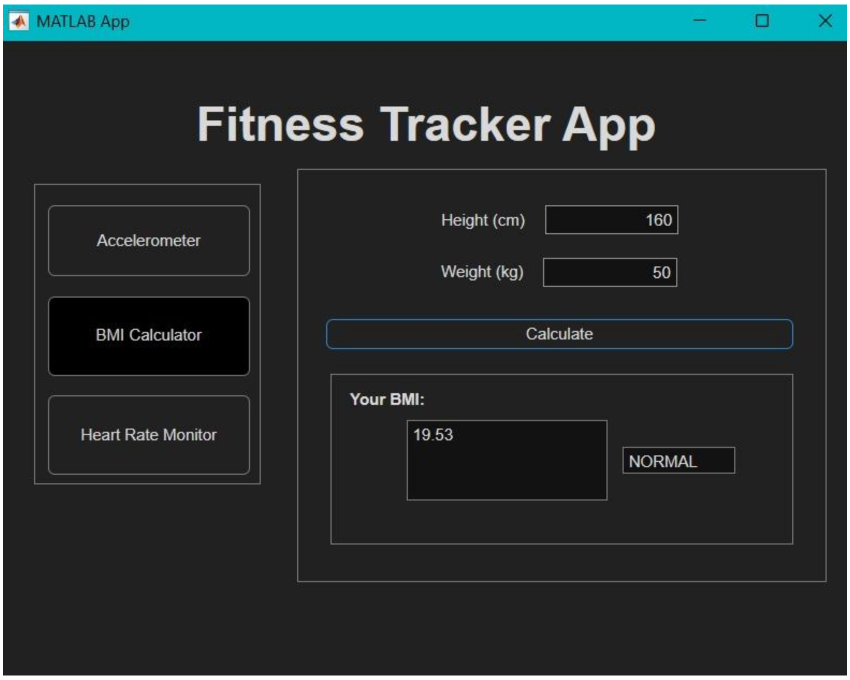
This module accesses the mobile device's accelerometer sensor to track movement or physical activity. It can be used to monitor steps, motion intensity, or posture, aiding in fitness tracking and activity logging.

BMI Calculator

The BMI (Body Mass Index) calculator lets users input height and weight to determine their BMI category (underweight, normal, overweight, etc.). It's a quick way to assess body composition and maintain a healthy weight range.

Heart Rate Monitor
This module reads and displays real-time heart rate data using a wave graph.

Wave Graph Explanation:
The graph shown displays a simulated heart rate pattern resembling a sinusoidal waveform. The X-axis represents time (in seconds), while the Y-axis shows amplitude (heartbeat intensity). This visual allows users to observe heartbeat regularity and patterns, which can help detect abnormalities or monitor intensity during workouts.



Height (cm) 160

Weight (kg) 50

Your BMI: 19.53 NORMAL

This image displays a "Fitness Tracker App" created in MATLAB, specifically showcasing its BMI (Body Mass Index) calculator functionality.

As you can see, the user interface is designed for straightforward interaction:
Input Fields: Users can easily input their Height (cm) and Weight (kg) into the designated text fields. In this example, the height is set to 160 cm and the weight to 50 kg.

Calculate Button: After entering the measurements, the user clicks the "Calculate" button (highlighted with a blue outline) to initiate the BMI calculation.

Results Display: Upon calculation, the app displays two key pieces of information in the lower section:

Your BMI: The calculated numerical BMI value (here, 19.53).

BMI Status: A qualitative assessment of the BMI, indicating whether it's "NORMAL," "Underweight," "Overweight," etc. In this case, 19.53 is categorized as "NORMAL."

This single-page layout provides a concise and user-friendly experience for quickly determining one's BMI and understanding its health implication. The app also features other functionalities like "Accelerometer" and "Heart Rate Monitor," accessible via the buttons on the left, demonstrating its comprehensive fitness tracking capabilities.

Conclusion

This fitness tracker demonstrates the power of MATLAB in developing robust, real-time health monitoring solutions. By integrating heart rate monitoring, accelerometer data, and BMI calculations, our project provides users with actionable insights to improve their fitness and overall well-being.