VELLORE INSTITUTE OF TECHNOLOGY

VIT

உழைப்பே உயர்வு தரும்

VIT®

BHOPAL

PROJECT REPORT

NAME - MOHIT YADAV

REG. NO. 25BAI11342

TOPIC -ElderHealthMonitoring&SOS Alert System (Python Simulation Project)

SUBJECT - CSE(1021)

FACULTY - VANDANA SHAKYA

Project Title

- ElderHealthMonitoring&SOS Alert System (Python Simulation Project)

• Project Overview:

- This project, built with Python, bringsto life a real-time health-monitoring system designed to help elderly individuals, especially those at risk of sudden medical emergencies. It simulates vital signs—like heart rate, body temperature, and oxygen saturation (SpO_2)—and checks these readings using realistic medical thresholds. If anything looks concerning, the system sends out an automatic SOS alert to signal for help. Because this project is
- written entirely in Python, there's no need for any special hardware or sensors. It's perfect for demonstrations, academic projects, or anyone wanting to understand essential programming concepts like breaking problems into modules, working with loops and functions, and building simulations.

• Real-World Problem Identified:

Many elderlypeoplelive withoutround-the-clocksupervision, which leads to:

Problems faced in reality

1. Delayed medical assistance if there are sudden increases in heart rate, oxygen levels, or fever.
2. No continuous monitoring of vitals when elderly individuals are alone.
3. Families uninformed in cases of health emergencies.
4. Lack of affordable medical alert systems for rural or middle-class families.
5. Delayed detection=high risk in case of heart attacks, strokes, and fever-related complications.

• Need of the System

- A simple, automated,continuous health-monitoring system can reduce these risks by the early detection of abnormal changes and the triggering of emergency alerts.

• Objectives

- Simulate real-time vital monitoring.
- Identify normal, warning, and critical health states.
- Trigger SOS alerts for critical conditions.
- Apply programming concepts like modular design, algorithms, and testing.
- Produce a working, easy-to-understand command-line application.

• Expected Outcomes

- Fully functional monitoring simulation.
- Clear and accurate detection of health conditions.
- SOS alerts that demonstrate emergency triggering logic.
- Proper logs and test results verifying program behavior.

• Structure of the Project

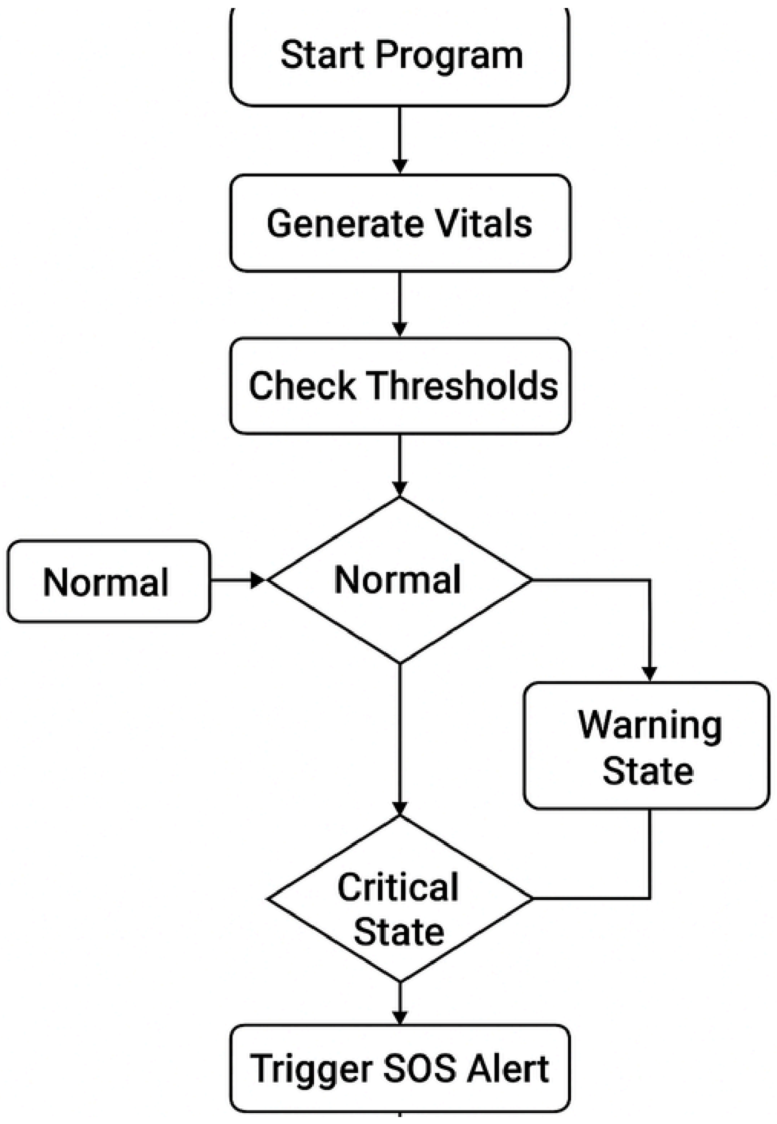
Main Program

- generate_vitals() → Simulates real-time vital data
- check_thresholds() → Classifies vitals into Normal / Warning / Critical
- log_reading() → (Optional) Saves data into CSV file
- trigger_sos() → Displaysemergencyalert
- main_loop() → Controls entire monitoring cycle

• Tools & Technologies Used

- **Python3**
- **random** → simulate vitals
- **time** →create delays
- **datetime** → timestamps for logs

• Flow Chart of the System



• Requirement Analysis

- Functional Requirements
  1. Simulate heart rate,temperature,andSpO_2.
  2. Compare with threshold values.
  3. Detect abnormal readings.
  4. Trigger alerts automatically.
- Non-Functional Requirements
  1. Reliability
  2. Usability
  3. Extensibility

4. Accuracy

• Algorithm

Step1: start the program.

Step2: Generate heart rate, temperature, and SpO_2 values.

Step3: Compare each value with threshold ranges.

```
Step4:   If normal   → Display "Normal"
         If warning  → Display "Warning"
         If critical → Trigger SOS
```

Step5: Log reading.

Step6: Repeat the loop.

• Implementation Details

- Thresholdsdefined as constants.
- Exception handling ensures no runtime crashes.
- Console messages formatted for clarity.
- Loops simulatecontinuous monitoring.
- Functions makecode modular and readable.

• Threshold Values

• Heart Rate (bpm)

- Normal: 60–100
- Warning: 101–120
- Critical: <40 or >120

• Temperature(°C)

- Normal: 36.1–37.2
- Warning: 37.3–38.9
- Critical: ≥39

• SpO_2 (%)

- Normal: ≥94%
- Warning: 90–93%
- Critical: <90%

• Testing&Detailed Test Cases

- Testing Methods

- Unit – Test each function independently Functional
  – Test the entire monitoring loop
- Stress
  – Run long-duration simulations
- Threshold
  – Manually inject critical reading

Problems Faced During Development

- **Random values changing too fast**
  → Solved byaddingtime.sleep().
- **False SOS**
  → Added refined threshold logic.
- **Repeated alerts**
  → Added conditional checks.
- **Difficult to test critical cases**
  → Forced test-modevalues.
- **Hard to maintain long code**
  → Usedmodularfunctions.

Refinement & Improvements Done

- improved console readability usingformattedstrings.
- Added morevitals for realistic simulation.
- Optimized threshold checks.
- Added optional CSV logging.
- Added delayto mimic real-time monitoring.
- Simplifiedfunctions forbetter gradingandreadability.

Future Enhancements

- Add IoT sensors like **ESP32/Arduino**.
- Real SMS/call alerts using **TwilioAPI**.
- Android appfor real-time display.
- Add ML-based health prediction model.
- Add GUI using Tkinter or React Native.

Conclusion

- TheElder Health Monitoring & SOS Alert System project illustrates that even a simple simulation based on Python can address a real, growing societal need. As the number of elderly grows, continuous health monitoring will become necessary to assure safety and timely medical help. This project shows how basic programming concepts in modular design, algorithms, and threshold-based decision-making can construct an effective, automated alert system. While the present edition uses simulated data, it forms a great groundwork for future integration with real sensors, mobile notifications, and IoT systems. With further refinement, this system can evolve into a practical tool making valuable contributions to elder care and emergency response.