**HACKVERESE 1.0**

# TEAM NAME: DIGITAL DEVELOPERS

HEALTH CARE

“CARE CONNECT – Automated Patient Alert System”

**PROBLEM STATEMENT:**

Here we create a real-time patient monitoring system that detects vital sign abnormalities and instantly alerts medical staff via calls or messages while securely storing and updating patient health data for continuous monitoring and analysis

We have got this idea while visiting the hospital randomly and we have discussed to create a app that can give alerts to respected/concerned doctors or nurses so they can immediately respond with in no time and can reduce the critical or serious emergencies of the particular patient.

We have followed a step-by-step process

**STEP 1:**

**->Objective:**

We have created a system to monitor vital signs like pulse rate, blood pressure, blood sugar etc.., post-operation and display them on mobile app.

We have to use both Hardware and Software in the prototype.

**STEP 2:** Selecting Hardware and Software Components

**->Hardware:**

Hardware components we use are different type of sensors to measure the vital sign changes the patient body.

Sensors:

Pulse Rate: Use a pulse oximeter sensor (e.g., MAX30102) to measure heart rate and SpO2 (oxygen saturation).

Blood Pressure: Non-invasive BP sensors (e.g., Omron or similar wearable BP modules). Alternatively, use a cuff-based sensor for periodic measurements.

Blood Sugar: Non-invasive glucose monitoring is challenging; consider integrating with existing devices like continuous glucose monitors (CGM) such as Dexcom or Freestyle Libre via APIs or Bluetooth.

Additional Sensors: Temperature sensor (e.g., DS18B20) or ECG sensor for heart activity (e.g., AD8232).

As there many sensors we have used a basic sensor to maintain a fast response.

**->Software:**

Software component we use is a mobile app to display real-time data, store historical data and alert users if vitals exceed safe thresholds

The main software tools use for Frontend are:

Html: To create a web page appearance,

CSS: To make the page more attractive.

TAILWIND:

TSX

TS

**->Connectivity:**

Wireless data transmission from sensors to the mobile app through wi-fi or Bluetooth. And it gives a notification for abnormal readings.

**->Microcontroller:**

Use a microcontroller like Arduino, Raspberry Pi, or ESP32 to process sensor data and handle connectivity.

**STEP 3:** Design the Hardware System

**->Circuit Design:**

We designed the circuit by connecting sensors to the microcontroller’s analog /digital pins. Ensure power management voltage regulators for stable sensor

We should use a breadboard or PCB to assemble the sensors, microcontroller, and connectivity module.

**STEP 4:** Develop the Software Backend

**->Data Processing:**

We used a Program for the microcontroller to read sensor data at regular intervals.

Implement algorithms to filter noise and ensure accurate measurements (e.g., averaging pulse readings over 10 seconds).