# College of Engineering and Technology

## Project Work – Student Hand Book

**Project Batch ID:** B506

| | | | |
|---|---|---|---|
| **Degree/ program** | B.Tech | **Specialisation** | Computer Science & Engineering |
| **Academic Year** | 2024-2025 (Even) | **Semester** | 6 |

| **Name of student** | **Register Number** | **Department** | **Mobile Number** | **Email ID** |
|---|---|---|---|---|
| VIVEK M G | RA2211003010002 | CTECH | 7904388983 | vm4512@srmist.edu.in |
| Raghuram Srikanth | RA2211003010218 | CTECH | 8610653380 | rs0657@srmist.edu.in |
| | | | | |
| | | | | |

| | |
|---|---|
| **Working Title of the Project:** | **Embedded Machine Learning for Early Detection of Heart Attack Symptoms** |
| **Project Site / Location** | SRM IST, Kattankulathur |
| **Name and address of the company / organisation (Applicable for projects with industry or industry support)** | SRM IST, Kattankulathur, Chengalpattu District-603203 |

**Supervision Team**

| | **Supervisor** | **Co-Supervisor** | **External Supervisor (If applicable)** |
|---|---|---|---|
| **Name** | Dr. R. Jeya | | |
| **Designation** | Associate Professor | | |
| **Department** | CTECH | | |
| **Campus** | Kattankulathur | | |
| **Telephone** | +91 9445360972 | | |
| **E-mail** | jeyar@srmist.edu.in | | |

| Course Code | 21CSP302L | Course Title | Project |
|---|---|---|---|

Mission Statement

To enhance driver safety by enabling real-time detection of early heart attack symptoms through embedded machine learning and physiological monitoring.

**Problem (or) Product Description:**

Heart attacks are a leading cause of sudden deaths and vehicle accidents, especially among long-distance drivers and individuals with undiagnosed heart conditions. Early symptoms often go unnoticed due to lack of immediate medical attention and limited self-awareness during transit. This project aims to address this challenge by developing an embedded system integrated into vehicles that continuously monitors vital physiological parameters — Heart Rate, $SpO_2$, Body Temperature, and Sweat Level.

Using a lightweight machine learning model deployed locally on the Arduino Nano 33 BLE Sense Rev2 board, the system can detect early signs of cardiac distress in real-time and trigger immediate alerts, enabling drivers to seek help before a serious incident occurs. The solution operates independently without requiring internet connectivity, making it ideal for remote or highway conditions where medical support is not immediately available.

**Assumptions and Constraints**

**Assumptions:**

- The driver will always wear or be in contact with the required biomedical sensors while driving.

- Environmental conditions (such as vehicle temperature) are within the operational tolerance of the sensors.
- The machine learning model can generalize well enough to detect anomalies across different users with minimal retraining.
- Emergency alerts (buzzer) will prompt the driver to take timely action.

**Constraints:**

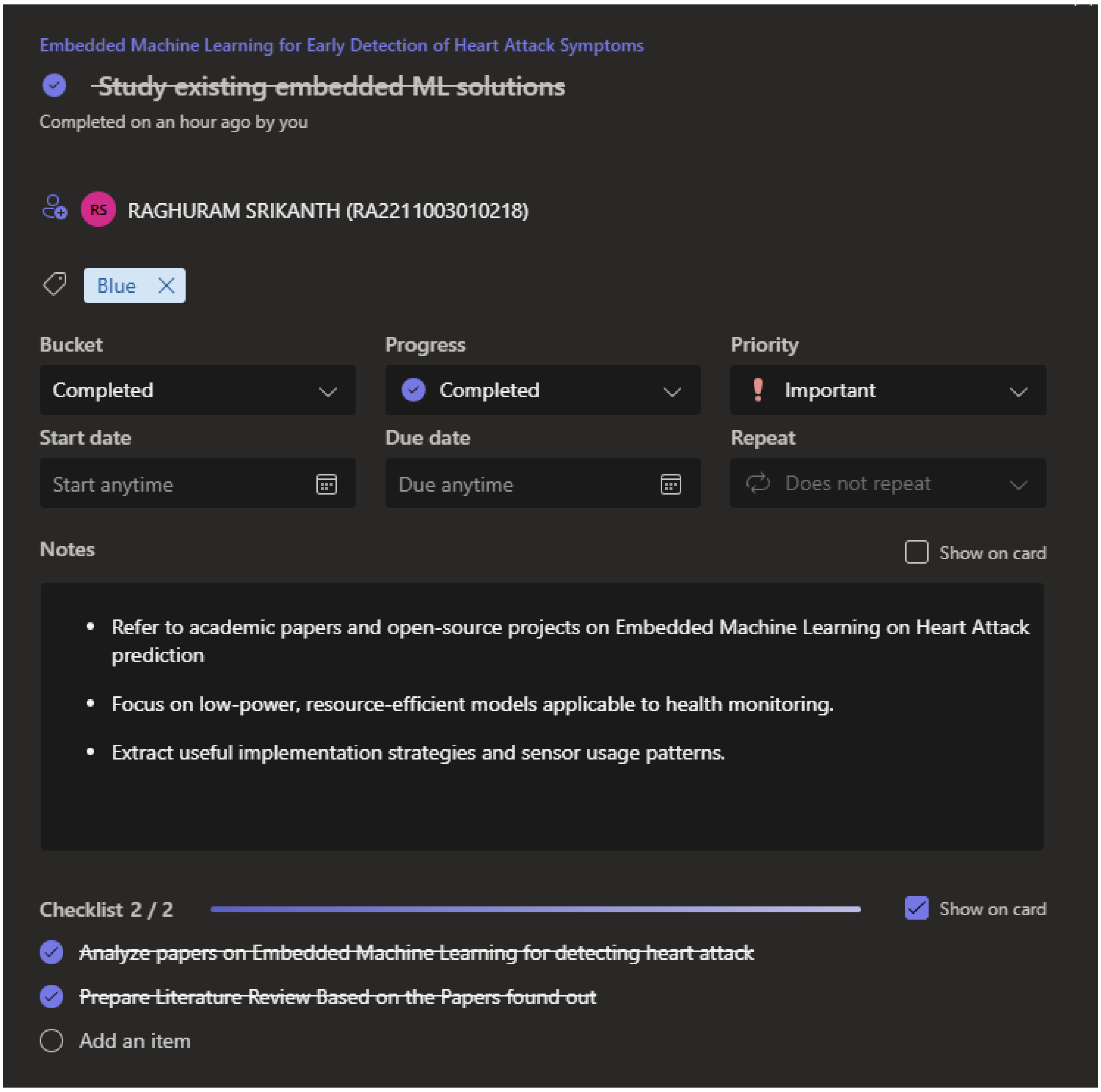
- Limited memory and processing capabilities of embedded hardware (Arduino Nano 33 BLE Sense Rev2).
- The system must operate with low power consumption for long drives without overheating or system failure.
- Limited sensor accuracy due to movement artifacts, vehicle vibration, or extreme environmental changes.
- Dataset used for model training may be synthetically balanced and may require retraining for field deployment.

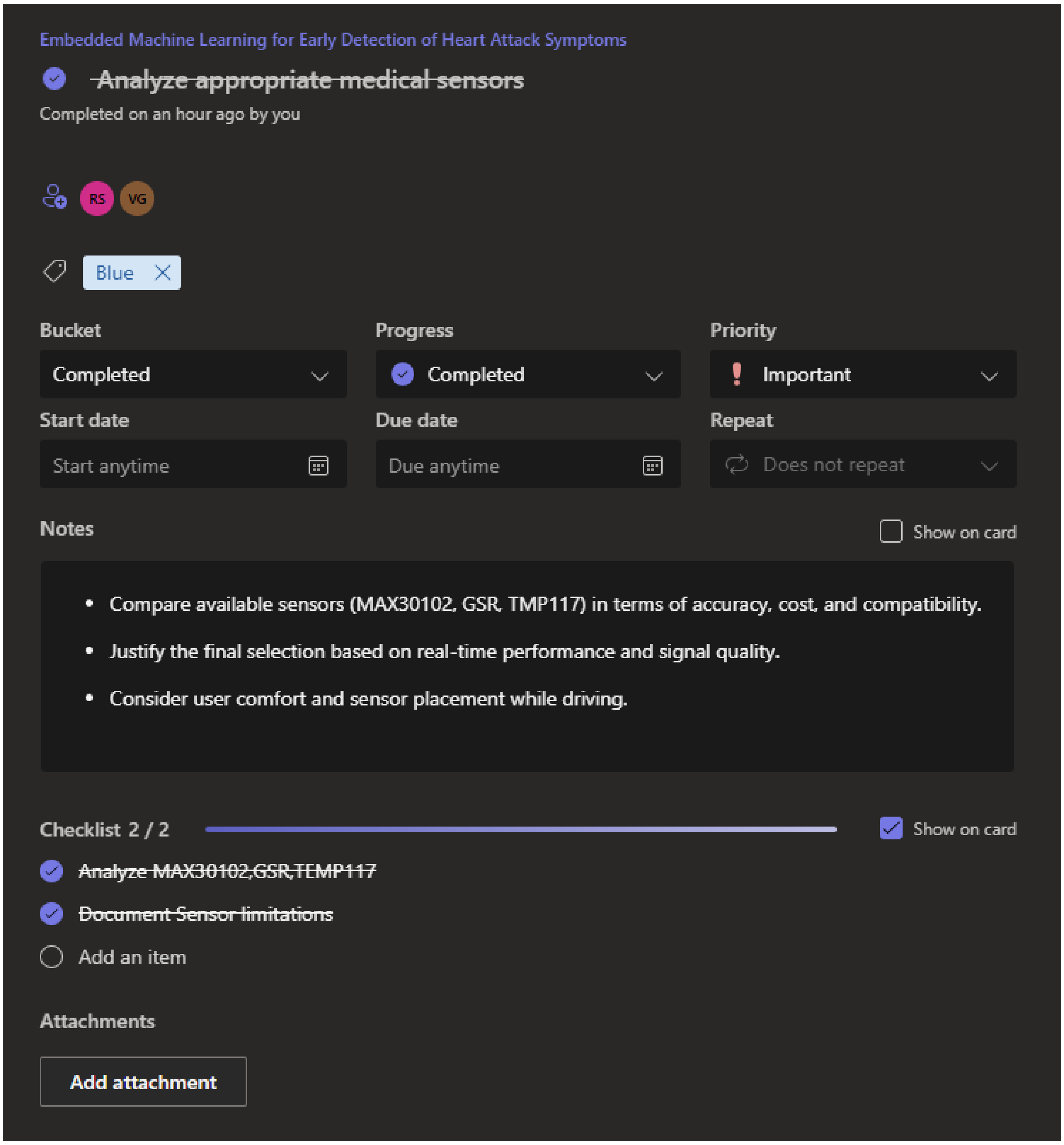
**Stakeholders**

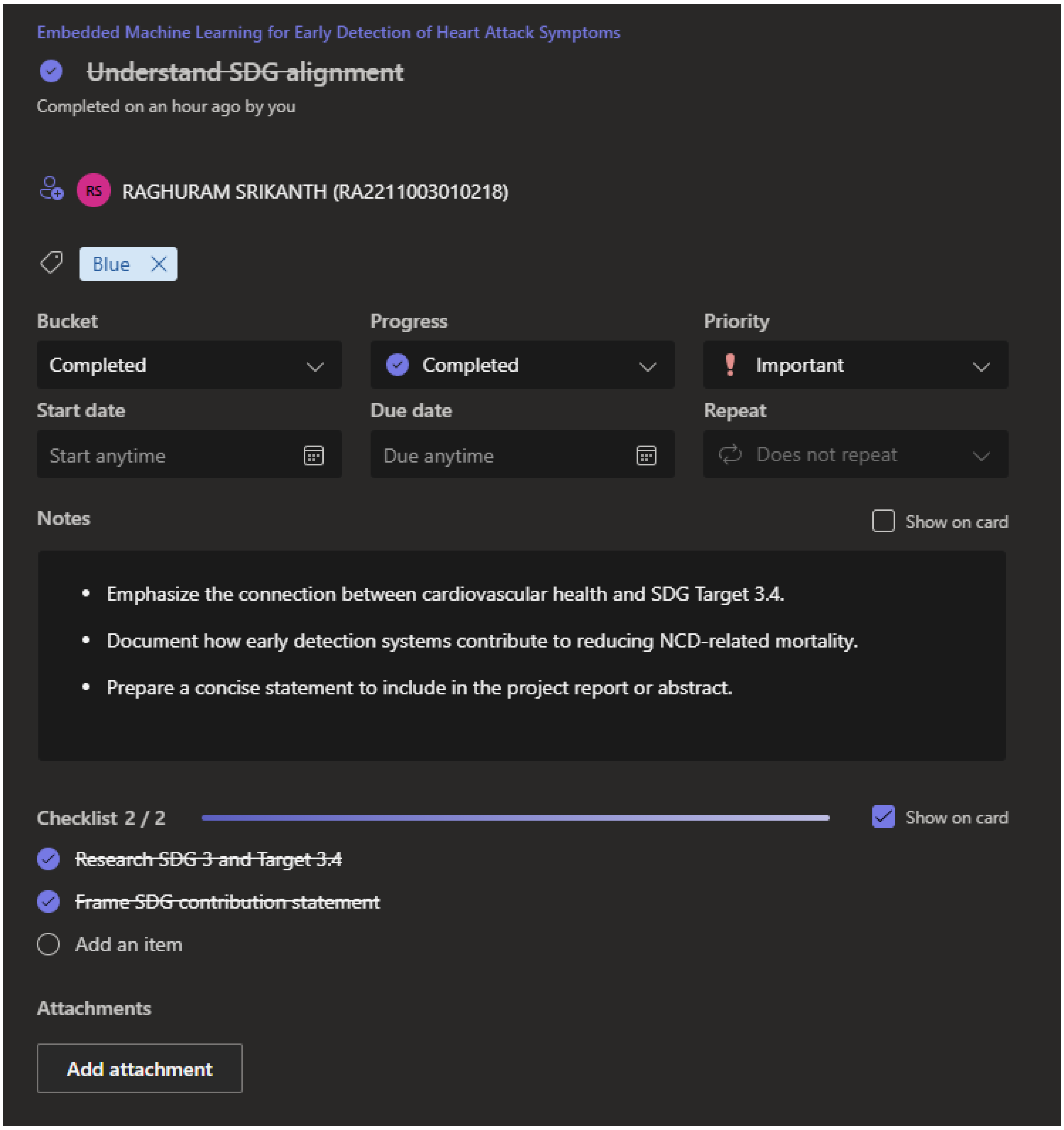
- Vehicle Drivers: Primary users who benefit directly from early detection and timely alerts.
- Automobile Manufacturers: Potential integrators of the system into vehicles for enhanced safety features.
- Healthcare Providers: Can use the system for remote monitoring and quick response support for high-risk individuals.

- Insurance Companies: May find interest in adopting such technology for risk assessment and accident prevention.
- Government and Road Safety Authorities: Stakeholders who can promote this technology to reduce road fatalities.
- Families of Drivers: Indirect beneficiaries through improved driver safety and emergency notification mechanisms.

**Division of work and contributors of SPRINT 1 [ Include Daily Scrum of Sprint 1]**




Embedded Machine Learning for Early Detection of Heart Attack Symptoms
~~Analyze appropriate medical sensors~~
Completed on an hour ago by you
RS
VG
Blue
Bucket
Completed
Progress
Completed
Priority
Important
Start date
Start anytime
Due date
Due anytime
Repeat
Does not repeat
Notes
Show on card
Compare available sensors (MAX30102, GSR, TMP117) in terms of accuracy, cost, and compatibility.
Justify the final selection based on real-time performance and signal quality.
Consider user comfort and sensor placement while driving.
Checklist 2 / 2
Show on card
~~Analyze MAX30102,GSR,TEMP117~~
~~Document Sensor limitations~~
Add an item
Attachments
Add attachment


Embedded Machine Learning for Early Detection of Heart Attack Symptoms
~~Understand SDG alignment~~
Completed on an hour ago by you
RS
RAGHURAM SRIKANTH (RA2211003010218)
Blue
Bucket
Completed
Progress
Completed
Priority
Important
Start date
Start anytime
Due date
Due anytime
Repeat
Does not repeat
Notes
Show on card
Emphasize the connection between cardiovascular health and SDG Target 3.4.
Document how early detection systems contribute to reducing NCD-related mortality.
Prepare a concise statement to include in the project report or abstract.
Checklist 2 / 2
Show on card
~~Research SDG 3 and Target 3.4~~
~~Frame SDG contribution statement~~
Add an item
Attachments
Add attachment

INITIAL WORK DONE

The Literature survey analysed 10 research papers focusing on Machine Learning Techniques for early heart attack detection. Common ML approaches were compared based on accuracy and efficiency.

DATASETS USED

The studies used various biomedical datasets including:

* MIT-BIH Arrhythmia Dataset – Widely used for ECG classification
* PhysioNet Challenge Datasets – Provides real-world biosignal data.
* UCI Heart Disease Dataset — Contains patient records with attributes like heart rate, cholestrol, and blood cholestrol.





* MIMIC-III Database — A large collection of hospital ICU data.

MACHINE LEARNING ALGORITHMS AND THEIR PERFORMANCE

| ALGORITHM | ACCURACY | NOTES |
|---|---|---|
| ANN (Artificial Neural Networks) | 85-95% | Good for pattern recognition in ECG signals. |
| SVM (Support Vector Machine) | 80%-90% | Effective with small datasets and high-dimensional data |
| Fuzzy Logic | 75-85% | Used for handling uncertainity in biosignal analysis. |
| Random Forest | 85-92% | High accuracy with structured datasets |

KEY FINDINGS FROM LITERATURE

1. Deep learning models (CNN, ANN) outperform traditional ML models in terms of accuracy but require more computational resources.

2. Feature selection techniques significantly impact model performance, making hybrid approaches effective.

3. Real time processing challenges exist due to hardware constraints in embedded systems.

4. SVM and Random Forest provide a good trade-off between accuracy and computational efficiency for embedded applications.





RELEVANCE TO OUR PROJECT

* Dataset Choice: We can use publicly available datasets like UCI or PhysioNet to pre-train our model before deploying it on live sensor data.

* Algorithm Selection: Given the limitations of hardware in Arduino Nano 33 BLE Sense Rev 2, Random Forest or SVM may be more suitable than deep learning models like CNN.

* Optimization Needs: The model must be compressed using TensorFlow Lite or Edge Impulse for deployment on embedded systems.

Hardware Components

1. Microcontroller Board
Arduino Nano 33 BLE Sense Rev 2
**Processor:** Nordic Semiconductor nRF52840 (ARM-Cortex-M4F, 64 MHz)
**Memory:** 1 MB Flash, 256 KB RAM
**Connectivity:** Bluetooth 5.0 low Energy

**Built-in Sensor:**
1. 9-axis IMU (LSM6DSOX, 3D accelerometer & gyroscope)
2. Environmental sensors: Temperature, humidity, barometric pressure
3. Microphone (MP34DT06JT) for audio-based ML applications

**Why we choose this microcontroller?**
- Optimized for TinyML with Tensorflow lite support
- Improved sensor accuracy and power efficiency compared to the previous version.
- Enhanced security features with NXP Edgelock SE050C secure element.





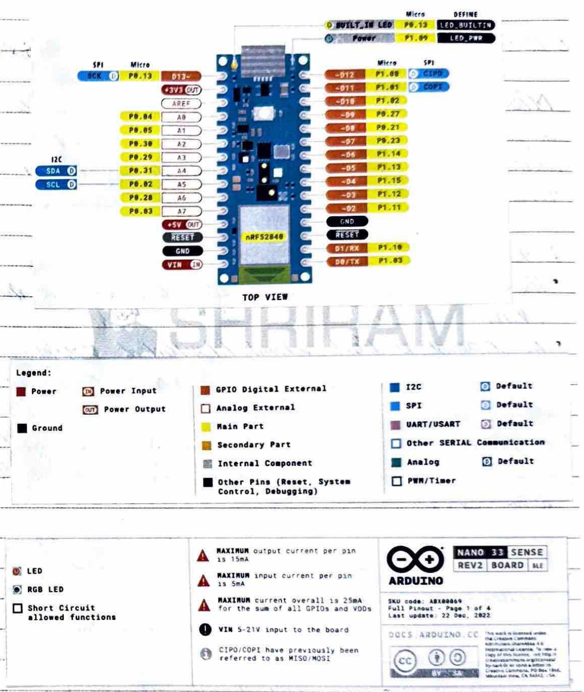



Fig: Pinlayout of the Microcontroller

## Sensor Overview:

### 1. MAX30102 - Pulse Oximeter and Heart Rate sensor

MAX30102 is an optical biosensor that integrates a pulse oximeter ($SPO_2$) and heart rate by detecting blood flow variations in the capillary tissue.

**Working Principle:**

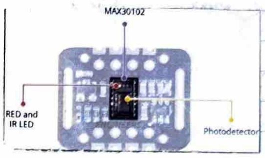
- The sensor emits red and infrared light into the skin.
- The amount of light absorbed/reflected varies depending on blood oxygen levels and heart rate (BPM) and $SPO_2$ (%)
- A photo ~~diat~~ detector measures the reflected light, and internal processing converts it into heart rate and SPO2

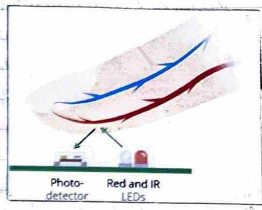
**Application in the Project**

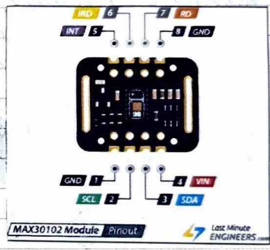
- Monitoring heart rate fluctuations
- Detecting oxygen saturation levels, which can indicate potential heart issues.
- Providing real-time physiological insights for anomaly detection.








### 2. TMP117 - High Precision Temperature Sensor

TMP117 is a high-accuracy digital temperature sensor with a precision of ±0.1 °C. It is used to measure body temperature, which is an important parameter for detecting stress, fever, or abnormal cardiovascular condition.

**Working Principle:**

- The sensor operates using I2C (Inter-Integrated

DATE :
Mo Tu We Th Fr Sa Su

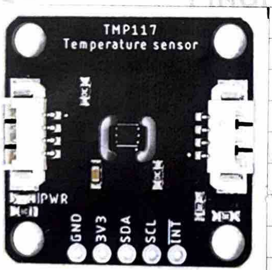


circuit) communication and provides high-resolution digital temperature reading

- It measures temperature changes in the skin and surrounding environment, helping in stress ~~[illegible]~~ detection and abnormal body conditions

Application in the project

- Monitoring body temperature variations, which can be an early indicator of cardiac distress
- Helping in correlating temperature changes with heart rate fluctuations.
- Assisting in thermal regulation studies for cardiovascular monitoring.






DATE :
Mo Tu We Th Fr Sa Su



3. Galvanic Skin Response Sensor (GSR)

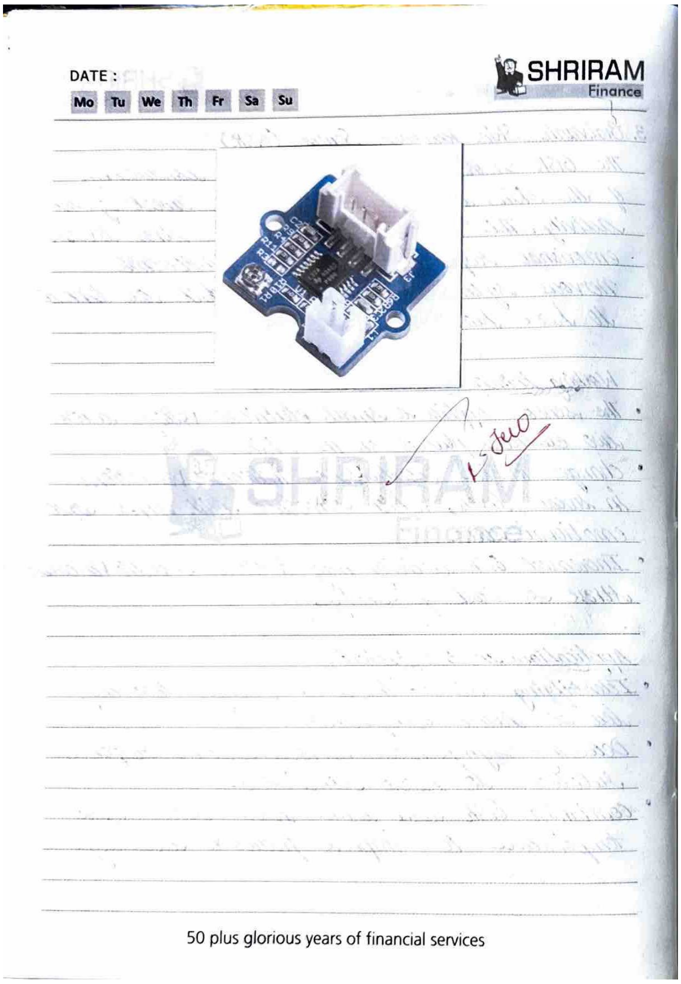
The GSR sensor measures the electrical conductivity of the skin, which changes based on sweat gland activity. This is useful for detecting stress levels, emotional responses, and ~~[illegible]~~ autonomic nervous system activity, all of which are linked to heart health.

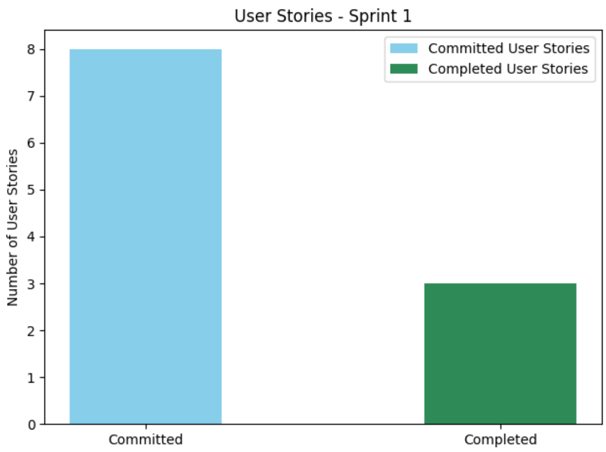
Working Principle:

- The sensor applies a small electrical voltage across two electrodes placed on the skin.
- Changes in skin conductance indicates variations in sweat levels, which correlate with stress and emotional state.
- Increased GSR activity may indicate cardiovascular stress or risk of arrhythmia

Applications in the Project:

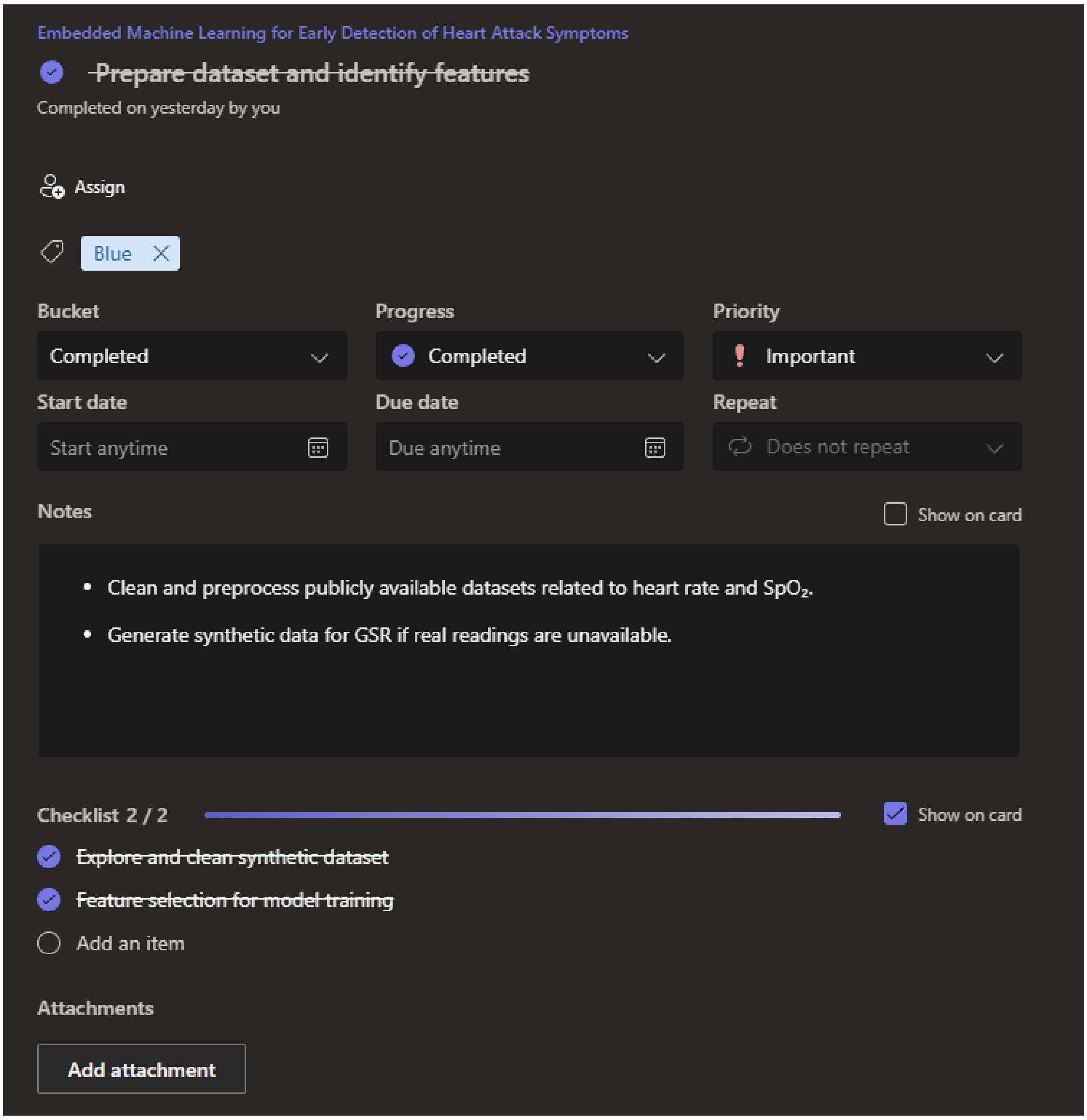
- Identifying stress-induced changes that may led to heart complication.
- Detecting sympathetic nervous system responses related to heart conditions.
- Combining GSR data with heart rate and temperature to improve prediction accuracy.

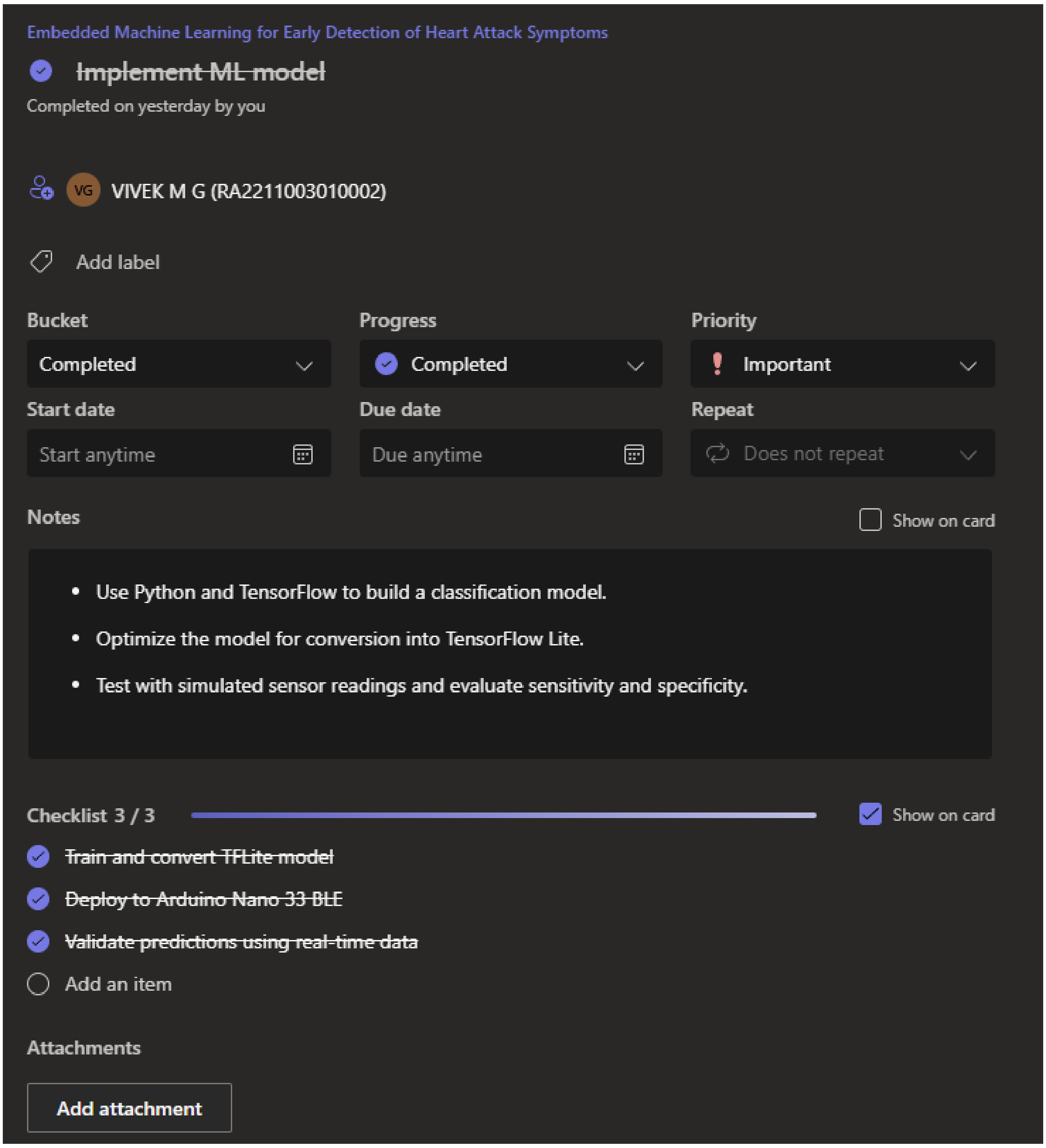



**Signature of the Supervisor**

**Division of work and contributors of SPRINT 2 [ Include Daily Scrum of SPRINT 2]**




Embedded Machine Learning for Early Detection of Heart Attack Symptoms
~~Implement ML model~~
Completed on yesterday by you
VG VIVEK M G (RA2211003010002)
Add label
Bucket
Completed
Progress
Completed
Priority
Important
Start date
Start anytime
Due date
Due anytime
Repeat
Does not repeat
Notes
Show on card
Use Python and TensorFlow to build a classification model.
Optimize the model for conversion into TensorFlow Lite.
Test with simulated sensor readings and evaluate sensitivity and specificity.
Checklist 3 / 3
Show on card
~~Train and convert TFLite model~~
~~Deploy to Arduino Nano 33 BLE~~
~~Validate predictions using real-time data~~
Add an item
Attachments
Add attachment

DATE :
Mo Tu We Th Fr Sa Su



Database Analysis :-
Pandas, numpy → data handling
matplotlib. pyplot, Seaborn → Visualization
sklearn. Preprocessing → for Scaling
tabulate → table display

Database Overview :-
Rows :- 50,000
Colums :- 5
Missing Value :- None
Feature Summary :-

| Feature | Type | Description |
|---|---|---|
| Heart_Rate | Float64 | Beats Per minute (38 to 150) |
| SpO2 | Float64 | Blood oxygen level in % |
| GSR | Float64 | Skin Conductance |
| Temperature | Float64 | Body Temperature in °C |
| Heart_Attack_Prediction | int64 | Target Variable |

DATE :
Mo Tu We Th Fr Sa Su



ML Implementation:-

1. Dataset:-
The dataset contains five columns:-
(i) Heart_Rate:- Heart Rate of individuals
(ii) SpO2:- Oxygen Saturation levels
(iii) GSR:- Galvanic Skin Response
(iv) Temperature:- Body Temperature
(v) Heart_Attack_Prediction:- The target variable.

2. Preprocessing:-
1. Feature and Target Seperation:
• X: Independent Variable (Heart Rate, SpO2, GSR, Temperature)
• Y: Dependent Variable (Heart_Attack_Prediction)

2. Scaling Features:-
• A MinMaxScaler was applied to scale features between 0 and 1. Scaling ensures that all features contribute equally

3. Train-Test Split:
• The data is split into training (80%) and testing (20%) subsets using train_test_split.



DATE :
Mo Tu We Th Fr Sa Su



3. Model Architecture:-
Layer Configuration:-
1. Input Layer:-
• 16 neurons with ReLU activation.
• Accepts 4 input features: Heart-Rate, SpO2, GSR and temperature
• Dropout rate of 30% to prevent overfitting

2. Hidden Layers:-
First Hidden Layer:- 8 neurons with ReLU activation + 30% dropout.
Second Hidden Layer: 4 neurons with ReLU activation.
3. Output Layer:-
~~Deep~~ Single Neuron with sigmoid activation for probability of date.
4. Training Process:-
(i) Configuration:-
• Optimizer: Adam with learning rate = 0.001
• Loss Function: Binary Cross entrophy
• Early stopping: Monitors Validation loss with Patience = 5 to Prevent overfitting
(ii) Training Parameters:-
• EPOCHS: 50
Batch size: 35

SHRIRAM Finance

DATE :
Mo Tu We Th Fr Sa Su

5. Performance Evaluation :-

Metrics :-

1. Test Accuracy: 96.64%
2. Precision/Recall:
    - Class 0 (No heart Attack): 99% Preci
    - Class 1 (Heart attack): 93% Precision, 98% recall
3. F1-Score: 97% weighted average

Loss Trends:-

1. Training loss decreased from 0.64 to 0.13
2. Validation loss stabilized at ~0.10, indicating effective generalization.

6. Deployment Preparation:-

- Converted to TensorFlow lite format for edge device deployment
- Model Size optimized for mobile/IOT applications while maintaining performance

7. Key strengths:-

1. Balanced Dataset: Equal class representation prevents bias
2. Regularization: Dropout layers + early stopping control overfitting effectively.
3. High Recall: 98% recall for heart attacks minimize false negatives.

50 plus glorious years of financial services

SHRIRAM Finance

DATE :
Mo Tu We Th Fr Sa Su

8. Results & Limitations :-

High Accuracy Justification :-

The model achieves 96.67% accuracy which may appear unusually high for medical prediction tasks. The

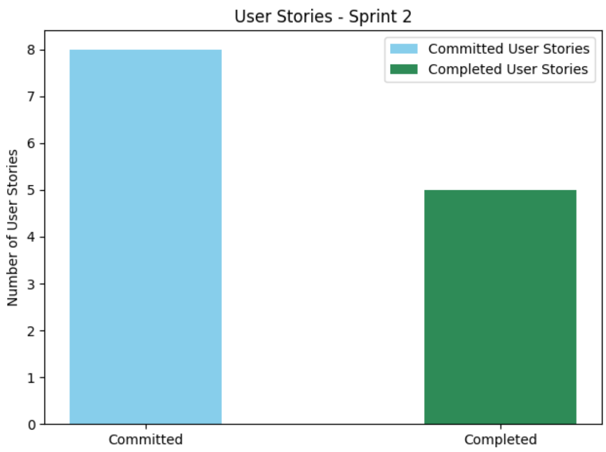
1. Synthetic Data characteristics :-

- The data was synthetically generated with strict thresholds for vital signs.
- Features were engineered to have clear decision boundaries, making classification easier than real world data.

2. Controlled Imbalance:-

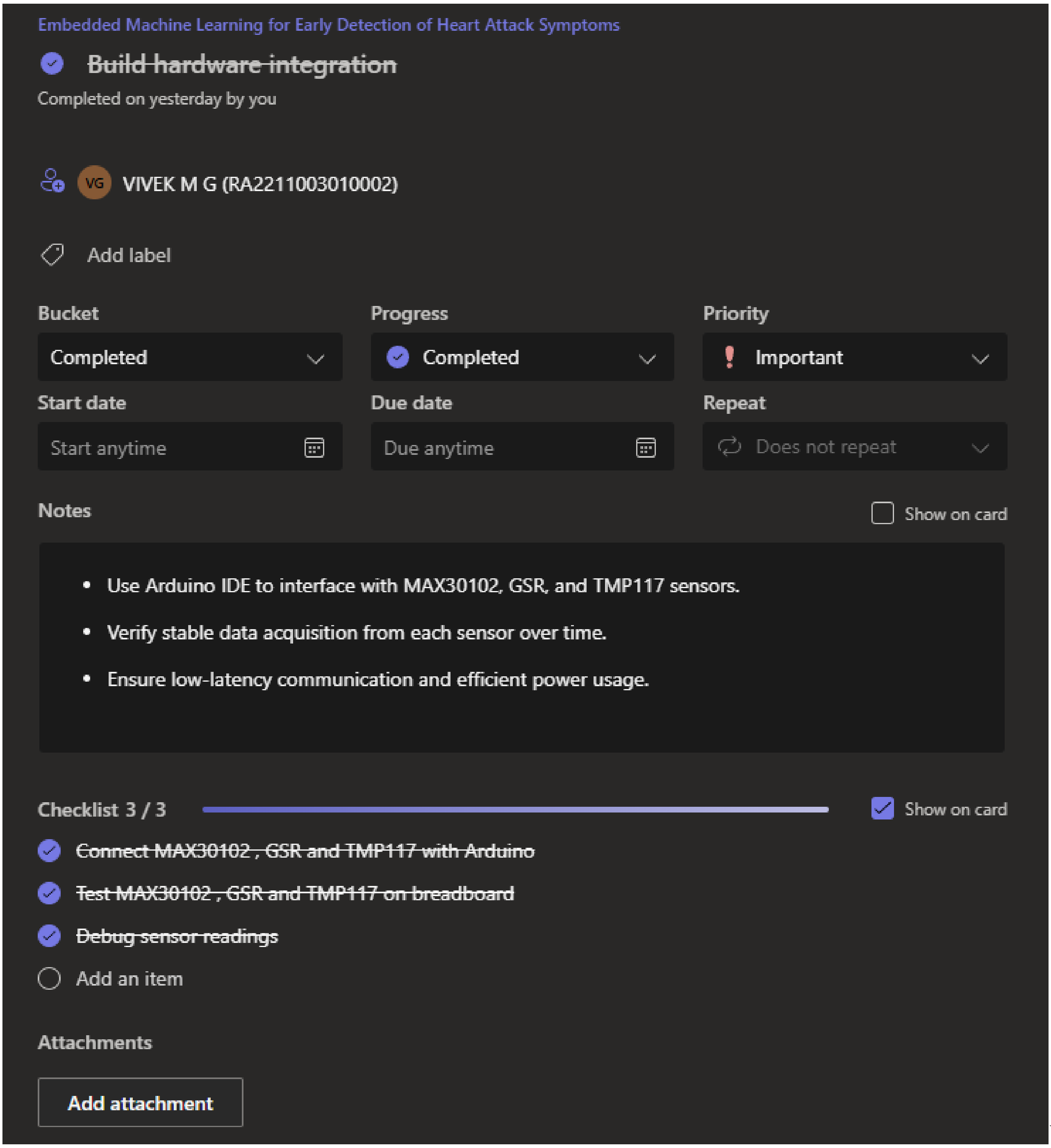
While labelled as 'balanced', the synthetic generation process inherently creates simplified patterns that reduce real world variability.

50 plus glorious years of financial services

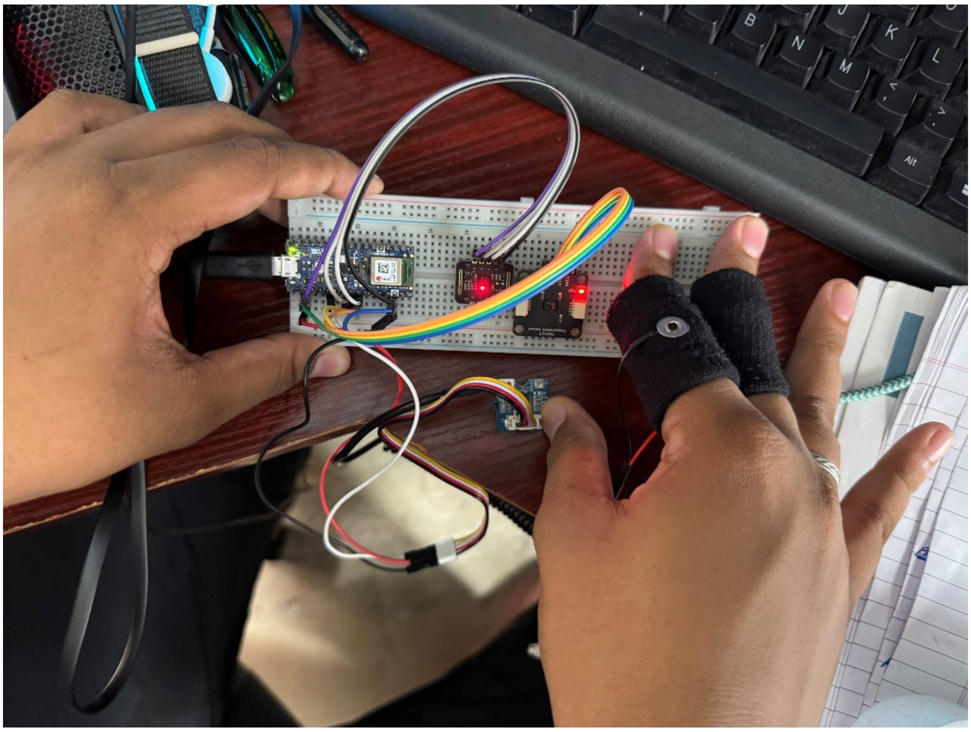



**Signature of the Supervisor**

**Division of work and contributors of SPRINT 3 [ Include Daily Scrum of Sprint 3]**




]

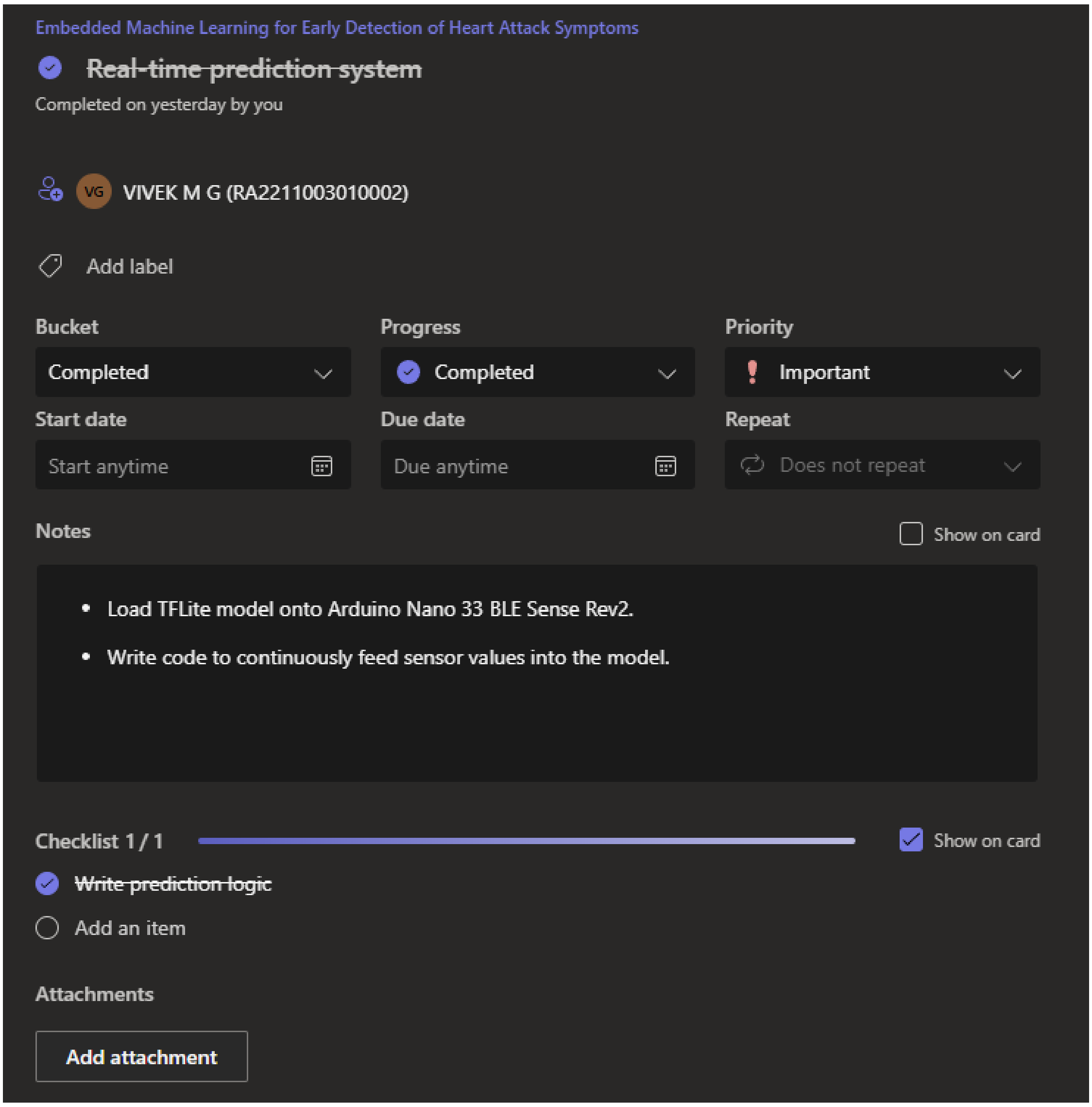


```
#include <Wire.h>
#include <TensorFlowLite.h>
#include <tensorflow/lite/micro/tflite_bridge/micro_error_reporter.h>
#include <tensorflow/lite/micro/micro_interpreter.h>
#include <tensorflow/lite/micro/all_ops_resolver.h>
#include <tensorflow/lite/schema/schema_generated.h>
#include <MAX30105.h>
#include <heartRate.h>
#include <spo2_algorithm.h>
#include <SparkFun_TMP117.h>
#include "model_data.h"  // Include trained TFLite model

// Sensor Initialization
MAX30105 particleSensor;
TMP117 tempSensor;
const int GSR_PIN = A0;

// TensorFlow Lite Variables
constexpr int tensorArenaSize = 20 * 1024;
uint8_t tensorArena[tensorArenaSize];

tflite::MicroErrorReporter errorReporter;
tflite::AllOpsResolver resolver;
```

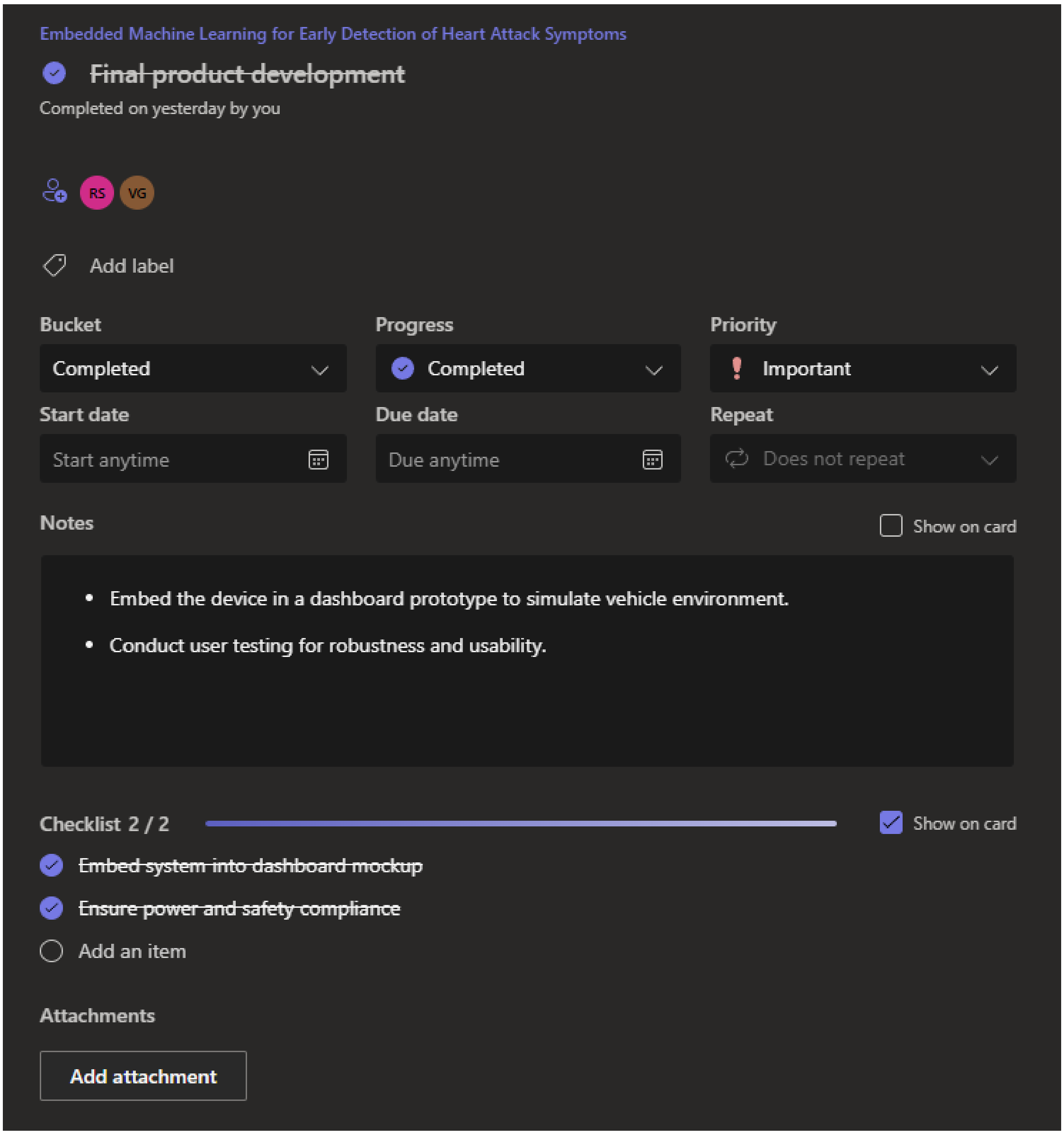
```
const tflite::Model* model;
tflite::MicroInterpreter* interpreter;

void setup() {
    Serial.begin(115200);
    Wire.begin();

    // Initialize MAX30102 Sensor
    if (!particleSensor.begin(Wire, I2C_SPEED_STANDARD)) {
        Serial.println("ERROR: MAX30102 Sensor Not Detected!");
        while (1);
    }
    particleSensor.setup();

    // Initialize TMP117 Sensor
    if (!tempSensor.begin()) {
        Serial.println("ERROR: TMP117 Sensor Not Detected!");
        while (1);
    }

    // Load TFLite Model
    model = tflite::GetModel(tflite_model);
    if (model->version() > TFLITE_SCHEMA_VERSION) {
        Serial.println("ERROR: Model schema version mismatch!");
        while (1);
    }

    static tflite::MicroInterpreter staticInterpreter(model, resolver, tensorArena,
tensorArenaSize);
    interpreter = &staticInterpreter;

    if (interpreter->AllocateTensors() != kTfLiteOk) {
        Serial.println("ERROR: Tensor allocation failed!");
        while (1);
    }
    Serial.println("TFLite Model Loaded Successfully.");
}

void loop() {
    uint32_t irBuffer[100], redBuffer[100];
    int32_t bufferLength = 100;
    int32_t heartRate = 0, spo2 = 0;
    int8_t validHeartRate = 0, validSpO2 = 0;
```

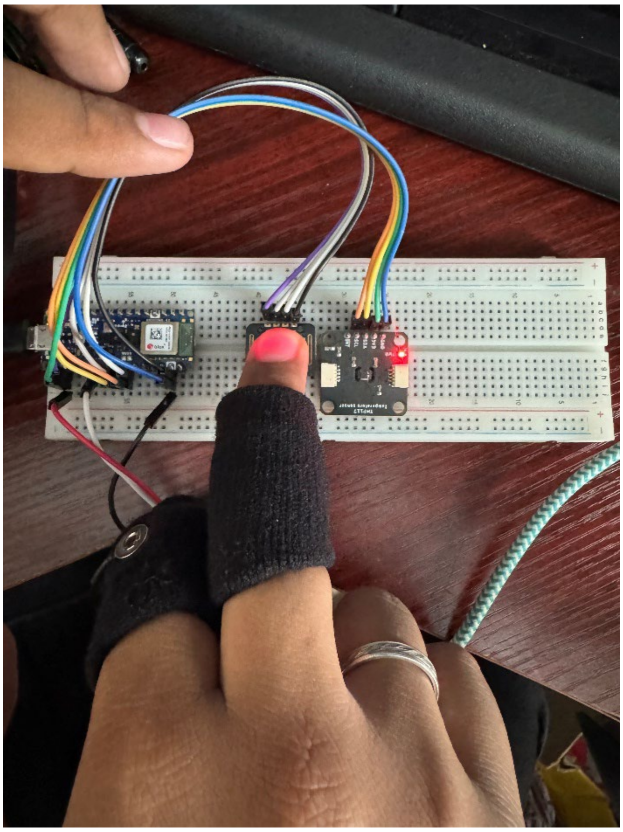
```
    // Read MAX30102 Sensor Data
    for (int i = 0; i < bufferLength; i++) {
        while (!particleSensor.available()) particleSensor.check();
        redBuffer[i] = particleSensor.getRed();
        irBuffer[i] = particleSensor.getIR();
        particleSensor.nextSample();
    }

    // Detect Finger Placement
    long irSignal = irBuffer[bufferLength - 1];
    bool fingerDetected = (irSignal > 5000);

    if (fingerDetected) {
        maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2,
&validSpO2, &heartRate, &validHeartRate);
    } else {
        heartRate = 0;
        spo2 = 0;
    }

    // Read TMP117 Temperature Sensor
    float temperature = fingerDetected ? tempSensor.readTempC() : 0;
    bool tempValid = (fingerDetected && !(isnan(temperature) || temperature < 10.0 ||
temperature > 50.0));
    if (!tempValid) temperature = 0;

    // Read GSR Sensor and Convert to Conductance
    int rawGsrValue = analogRead(GSR_PIN);
    float gsrValue = (rawGsrValue >= 1000 || rawGsrValue == 0) ? 0.0 : (1.0 / ((3.3 *
1000000.0 / ((float)rawGsrValue * (3.3 / 1023))) - 1000000.0)) * 1000000.0;

    // Prepare TensorFlow Lite Input
    TfLiteTensor* input = interpreter->input(0);
    if (input->type == kTfLiteFloat32 && input->dims->size == 2 && input->dims->data[1]
== 4) {
        input->data.f[0] = (float)heartRate;
        input->data.f[1] = (float)spo2;
        input->data.f[2] = temperature;
        input->data.f[3] = gsrValue;
    } else {
        Serial.println("ERROR: Input tensor type mismatch!");
        return;
```

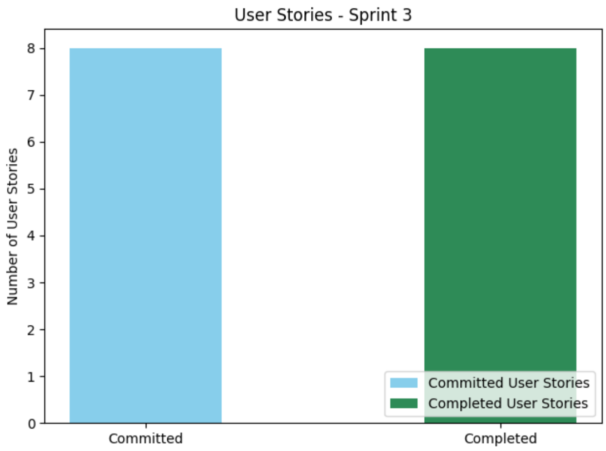
```
    }

    // Run Inference
    if (interpreter->Invoke() != kTfLiteOk) {
        Serial.println("ERROR: TFLite inference failed!");
        return;
    }

    // Get Output Tensor
    TfLiteTensor* output = interpreter->output(0);
    float heartAttackRisk = fingerDetected ? output->data.f[0] : 0.0;

    // Print Results
    Serial.print("HR: "); Serial.print(validHeartRate ? heartRate : 0);
    Serial.print(" bpm | SpO2: "); Serial.print(validSpO2 ? spo2 : 0);
    Serial.print("% | Temp: "); Serial.print(tempValid ? temperature : 0);
    Serial.print(" °C | GSR: "); Serial.print(gsrValue);
    Serial.print(" | Heart Attack Risk: "); Serial.println(heartAttackRisk);

    delay(5000);
}
```




```
HR: 187 bpm | SpO2: 12% | Temp: 31.03 °C | GSR: 0.95 | Heart Attack Risk: 0.03
HR: 166 bpm | SpO2: 100% | Temp: 31.09 °C | GSR: 1.17 | Heart Attack Risk: 0.12
HR: 107 bpm | SpO2: 96% | Temp: 31.12 °C | GSR: 0.63 | Heart Attack Risk: 0.00
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.42 | Heart Attack Risk: 0.00
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.82 | Heart Attack Risk: 0.00
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.81 | Heart Attack Risk: 0.00
HR: 100 bpm | SpO2: 0% | Temp: 31.26 °C | GSR: 0.54 | Heart Attack Risk: 1.00
HR: 136 bpm | SpO2: 92% | Temp: 31.30 °C | GSR: 0.37 | Heart Attack Risk: 0.04
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.29 | Heart Attack Risk: 0.00
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.28 | Heart Attack Risk: 0.00
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.37 | Heart Attack Risk: 0.00
HR: 0 bpm | SpO2: 0% | Temp: 0.00 °C | GSR: 0.45 | Heart Attack Risk: 0.00
HR: 71 bpm | SpO2: 98% | Temp: 31.49 °C | GSR: 0.42 | Heart Attack Risk: 0.00
```


Embedded Machine Learning for Early Detection of Heart Attack Symptoms
~~Final product development~~
Completed on yesterday by you
RS
VG
Add label
Bucket
Completed
Progress
Completed
Priority
Important
Start date
Start anytime
Due date
Due anytime
Repeat
Does not repeat
Notes
Show on card
Embed the device in a dashboard prototype to simulate vehicle environment.
Conduct user testing for robustness and usability.
Checklist 2 / 2
Show on card
~~Embed system into dashboard mockup~~
~~Ensure power and safety compliance~~
Add an item
Attachments
Add attachment






**Signature of the Supervisor**

**Worksheet / Data collection / Observation, etc**

| Heart_Rate | SpO2 | GSR | Temperature | Heart_Attack_Prediction |
|---|---|---|---|---|
| 79.96714 | 97.7926 | 13.94307 | 37.26151 | 0 |
| 97.396 | 92.53173 | 24.31685 | 38.52818 | 1 |
| 70.30526 | 98.81384 | 10.60975 | 36.26714 | 0 |
| 109.0221 | 89.17344 | 11.10033 | 37.1064 | 0 |
| 70.71954 | 96.88639 | 9.889969 | 35.43019 | 0 |
| 92.9108 | 94.19544 | 25 | 38.15823 | 1 |
| 68.99746 | 99.42116 | 12.8731 | 36.18222 | 0 |
| 96.83458 | 92.67649 | 15.8654 | 37.77376 | 1 |
| 62.34881 | 99.63799 | 20.33494 | 37.09682 | 0 |
| 105.3372 | 94.76352 | 13.96366 | 36.39169 | 1 |
| 70.39361 | 99.58568 | 13.03085 | 35.61848 | 0 |
| 103.9442 | 92.22984 | 15.29231 | 38.37281 | 1 |
| 85.31 | 99.39692 | 9.482347 | 36.34539 | 0 |
| 107.085 | 94.95109 | 16.0833 | 37.66759 | 1 |
| 83.12526 | 100.0344 | 11.78397 | 37.00177 | 0 |
| 112.0493 | 91.70976 | 19.44558 | 38.92476 | 1 |

**Research Article with Journal Publication Details / Patent disclosure form with patent status**

(include certificates and proofs)