Final Deliverable

code

Team ID	PNT2022TMID45391
Project Name	Project-Industry-specific intelligent fire management system

Submitted by:
Team Leader: ARUL B
Team member: AHAMED HABEEB ANSARI V U
Team member: ARUN KUMAR R
Team member: YUVARAJA E

Code:

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);  //define I2C address 0x27, 16 column and 2 rows

float flamelevel = 0; // mapped and inverted % of sensor range
#define greenLED 7 // fire okay or firing
#define redLED 2 // warning or flashing alert

// empirically relate flame % to actual fire condition (TBD)
const int minSurvive = 15; // minimum level for idle, below is outfire
const int idleLow = 20; // lowest reading for healthy idle
const int idleTarget = 30; // target reading for resting idle
const int firingLow = 70; // lowest reading for actively firing
const int firingHigh = 90; // reading for full firing



void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();
  Serial.begin(9600);

  pinMode(greenLED, OUTPUT); // set green pin led as output
  digitalWrite(greenLED, LOW); // turn off green led

  pinMode(redLED, OUTPUT); // set red led pin as output
  digitalWrite(redLED, LOW); // turn off red led
}

void loop() {
```

```
float analogValue = analogRead(A0);
Serial.print("Sensor RAW: ");
Serial.println(analogValue, 0);
flamelevel = map(analogValue, 0, 1024, 100, 0);
Serial.print(flamelevel, 0);
Serial.println("%");

// disabling the lcd commands makes serial print work
lcd.setCursor(0, 0);
lcd.print(F("Flame: "));

if (flamelevel >= firingHigh) { // stoker is fully firing
  lcd.print("Full Fire");
  digitalWrite(greenLED, HIGH); // turn on green led
  digitalWrite(redLED, LOW); // turn off red led
  delay(300);
  digitalWrite(greenLED, LOW); // turn off green led for flash
}

if ((flamelevel >= firingLow) && (flamelevel < firingHigh)) { // stoker is firing
  lcd.print("Firing   ");
  digitalWrite(greenLED, HIGH); // turn on green led
  digitalWrite(redLED, LOW); // turn off red led
}

if ((flamelevel < firingLow) && (flamelevel > idleLow) ) {  // idle fire
  lcd.print("Idle fire ");
  digitalWrite(greenLED, HIGH); // turn on green led
  digitalWrite(redLED, HIGH); // turn off red led
}

if ((flamelevel <= idleLow) && (flamelevel >= minSurvive) ) {  // low fire
  lcd.print("Low fire  ");
  digitalWrite(greenLED, LOW); // turn on green led
  digitalWrite(redLED, HIGH); // turn off red led
  // trigger stoker run timer = 2 mins?
}

if (flamelevel < minSurvive) {  // fire out
  lcd.print("FIRE OUT! ");
  digitalWrite(greenLED, LOW); // turn on green led
  digitalWrite(redLED, HIGH); // turn off red redLED
  delay(300);
  digitalWrite(redLED, LOW); // turn off red led for flash
  // send alert
}
```

```
  lcd.setCursor(0, 1);
  lcd.print("   Level: ");
  lcd.print(flamelevel, 0);
  lcd.print("%    ");

  delay(200);
}
```