```
#include <Wire.h>
#include <U8g2lib.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <WiFiManager.h>
#include <HTTPClient.h>
#include <FirebaseESP32.h>
#include <SD.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDRESS 0x3C
#define RXD2 16
#define TXD2 17
#define BUTTON_PIN 2

TinyGPSPlus gps;
SoftwareSerial ss(RXD2, TXD2);
U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE, SCL, SDA);

bool wifiConnected = false;
bool wifiButtonPressed = false;
bool isChallanGenerated = false;
unsigned long challanDisplayStartTime = 0;
int alertCount = 0;
const char* serverUrl = "https://challan.glitch.me";
const int chipSelect = 5;

// Firebase configuration
#define DATABASE_URL "https://your-firebase-database-url.firebaseio.com/"
#define DATABASE_SECRET "your-firebase-secret"
```

```
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

void setup() {
  Serial.begin(115200);
  ss.begin(9600);
  u8g2.begin();
  u8g2.clearBuffer();
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  initFirebase();

  if (!SD.begin(chipSelect)) {
    Serial.println("SD card initialization failed!");
    return;
  }
  Serial.println("SD card initialized.");
}

void loop() {
  boolean newData = false;
  for (unsigned long start = millis(); millis() - start < 1000;) {
    while (ss.available()) {
      if (gps.encode(ss.read())) {
        newData = true;
      }
    }
  }
  if (newData) {
    newData = false;
```

```
    drawDisplay();
    float speed = getSpeed();
    sendSpeedDataToFirebase(speed);
  } else {
    drawNoData();
  }
  if (digitalRead(BUTTON_PIN) == LOW) {
    wifiButtonPressed = true;
  }
  if (wifiButtonPressed && !wifiConnected) {
    connectToWiFi();
  }
  delay(1000);
}

void connectToWiFi() {
  WiFiManager wifiManager;
  wifiManager.autoConnect("AutoConnectAP");
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("WiFi connected!");
    wifiConnected = true;
  } else {
    Serial.println("WiFi connection failed!");
    wifiConnected = false;
  }
}

void initFirebase() {
  config.database_url = DATABASE_URL;
  config.signer.tokens.legacy_token = DATABASE_SECRET;
  Firebase.begin(&config, &auth);
```

```
}

float getSpeed() {
  float speed = gps.speed.kmph();
  return speed;
}

void sendSpeedDataToFirebase(float speed) {
  if (Firebase.setFloat(fbdo, "/speed", speed)) {
    Serial.println("Speed data sent to Firebase");
  } else {
    Serial.println("Failed to send speed data to Firebase");
  }

  if (Firebase.setFloat(fbdo, "/latitude", gps.location.lat())) {
    Serial.println("Latitude data sent to Firebase");
  } else {
    Serial.println("Failed to send latitude data to Firebase");
  }

  if (Firebase.setFloat(fbdo, "/longitude", gps.location.lng())) {
    Serial.println("Longitude data sent to Firebase");
  } else {
    Serial.println("Failed to send longitude data to Firebase");
  }
}

void drawDisplay() {
  u8g2.clearBuffer();
  drawSpeed();
  drawSecondPart();
```

```
  u8g2.sendBuffer();
}

void drawSpeed() {
  u8g2.setFont(u8g2_font_t0_12_mr);
  u8g2.setCursor(0, 10);
  u8g2.print("SPEED");
  u8g2.setFont(u8g2_font_helvR24_tf);
  u8g2.setCursor(10, 42);
  float currentSpeed = gps.speed.kmph();
  u8g2.print(currentSpeed, 0);
  u8g2.drawVLine(45, 0, u8g2.getDisplayHeight());
  u8g2.setFont(u8g2_font_helvB08_tf);
  u8g2.setCursor(50, 60);
  u8g2.print("L 80Km | A ");
  u8g2.print(alertCount);
  u8g2.print("/10");
  if (alertCount >= 10) {
    isChallanGenerated = true;
    challanDisplayStartTime = millis();
    u8g2.setCursor(50, 40);
    u8g2.print("Challan Generated");
    sendChallanInfo();
  }
}

void drawSecondPart() {
  u8g2.setFont(u8g2_font_t0_12_mr);
  u8g2.setCursor(50, 10);
  if (wifiConnected) {
    u8g2.print("WiFi Connected");
```

```
  } else {
    u8g2.print("WiFi Not Connected");
  }
}

void drawNoData() {
  u8g2.clearBuffer();
  u8g2.setFont(u8g2_font_t0_12_mr);
  u8g2.setCursor(0, 10);
  u8g2.print("No Data");
  u8g2.sendBuffer();
}

void sendChallanInfo() {
  HTTPClient http;
  String postBody = "{\"latitude\": " + String(gps.location.lat(), 6) +
                    ", \"longitude\": " + String(gps.location.lng(), 6) +
                    ", \"time\": \"" + String(gps.time.value()) + "\"" +
                    ", \"challanGenerationInfo\": \"Info\"" +
                    ", \"espId\": \"ESP32\"" +
                    ", \"challanCount\": " + String(alertCount) + "}";
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/json");
  int httpResponseCode = http.POST(postBody);
  if (httpResponseCode > 0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
  } else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
```

```
  http.end();

  // Save challan information to SD card
  File dataFile = SD.open("challan_info.txt", FILE_WRITE);
  if (dataFile) {
    dataFile.println("Latitude: " + String(gps.location.lat(), 6));
    dataFile.println("Longitude: " + String(gps.location.lng(), 6));
    dataFile.println("Time: " + String(gps.time.value()));
    dataFile.println("Challan Generation Info: Info");
    dataFile.println("ESP ID: ESP32");
    dataFile.println("Challan Count: " + String(alertCount));
    dataFile.println("-------------------------------------");
    dataFile.close();
    Serial.println("Challan info saved to SD card.");
  } else {
    Serial.println("Error opening file for writing.");
  }
}
```