Smart Biometric Access System using ESP32

Code:

```
#include <Adafruit_Fingerprint.h>
#include <WiFi.h>
#include <WebServer.h>
#include "FS.h"
#include "SPIFFS.h"
#include <time.h>

#define R307_RX 16
#define R307_TX 17
#define LED_PIN 14
#define BUZZER_PIN 15

const char* ssid = "shreyankismad";
const char* password = "12345678";

WebServer server(80);
bool isLoggedIn = false;

const String loginUser1 = "admin";
const String loginPass1 = "1234";
const String loginUser2 = "guest";
const String loginPass2 = "5678";

HardwareSerial mySerial(2);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

struct User {
  int fingerID;
  String name;
};

User users[] = {
  {1, "Rahul"}
};
int userCount = sizeof(users) / sizeof(users[0]);

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);

  mySerial.begin(57600, SERIAL_8N1, R307_RX, R307_TX);
  finger.begin(57600);

  if (finger.verifyPassword()) {
```

```
  Serial.println("R307 Fingerprint sensor detected");
} else {
  Serial.println("Fingerprint sensor not found");
  while (1) delay(1);
}

if (!SPIFFS.begin(true)) {
  Serial.println("SPIFFS Mount Failed");
} else {
  Serial.println("SPIFFS Initialized");
}

WiFi.begin(ssid, password);
Serial.print("Connecting to Wi-Fi");
int tries = 0;
while (WiFi.status() != WL_CONNECTED && tries < 20) {
  delay(500);
  Serial.print(".");
  tries++;
}

if (WiFi.status() == WL_CONNECTED) {
  Serial.println("\nConnected to Wi-Fi");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  configTime(19800, 0, "pool.ntp.org");
```

```
    struct tm timeinfo;
    while (!getLocalTime(&timeinfo)) {
      delay(1000);
      Serial.print(".");
    }
    Serial.println("\nTime synchronized");
  } else {
    Serial.println("\nFailed to connect to Wi-Fi");
  }

  server.on("/", HTTP_GET, []() {
    if (isLoggedIn) {
      server.send(200, "text/html", "<h1>Welcome to Biometric Dashboard</h1><p><a href='/logs'>View Logs</a> | <a href='/logout'>Logout</a></p>");
    } else {
      server.send(200, "text/html",
        "<form action='/login' method='POST'>"
        "Username: <input type='text' name='user'><br>"
        "Password: <input type='password' name='pass'><br>"
        "<input type='submit' value='Login'>"
        "</form>");
    }
  });

  server.on("/login", HTTP_POST, []() {
    String user = server.arg("user");
    String pass = server.arg("pass");
    if ((user == loginUser1 && pass == loginPass1) || (user == loginUser2 && pass == loginPass2)) {
      isLoggedIn = true;
      server.sendHeader("Location", "/");
      server.send(303);
    } else {
      server.send(401, "text/html", "Login Failed. <a href='/'>Try Again</a>");
    }
  });

  server.on("/logout", HTTP_GET, []() {
    isLoggedIn = false;
    server.sendHeader("Location", "/");
    server.send(303);
  });

  server.on("/logs", HTTP_GET, []() {
    if (!isLoggedIn) {
      server.send(401, "text/plain", "Unauthorized. Please login first.");
      return;
    }
    File file = SPIFFS.open("/access_log.txt", "r");
    if (!file) {
      server.send(500, "text/plain", "Failed to open log file.");
      return;
    }
    String content = "<h2>Access Log</h2><pre>";
    while (file.available()) {
      content += (char)file.read();
    }
    content += "</pre><p><a href='/'>Back</a></p>";
    file.close();
    server.send(200, "text/html", content);
  });

  server.begin();
}

void loop() {
  server.handleClient();
  checkFingerprint();
}

void checkFingerprint() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return;

  p = finger.fingerFastSearch();
```

```
if (p == FINGERPRINT_OK) {
  int fingerID = finger.fingerID;
  String name = "Unknown";

  for (int i = 0; i < userCount; i++) {
    if (users[i].fingerID == fingerID) {
      name = users[i].name;
      break;
    }
  }

  String logEntry = "Fingerprint ID: " + String(fingerID);
  logEntry += " Access Granted\n";
  logEntry += "Name: " + name + "\n";
  logEntry += "Time: " + getCurrentTime() + "\n";
  Serial.print(logEntry);
  logToSPIFFS(logEntry);

  digitalWrite(LED_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH);
  delay(200);
  digitalWrite(BUZZER_PIN, LOW);
  delay(300);
  digitalWrite(LED_PIN, LOW);
} else {
  String logEntry = "Fingerprint Scan Failed\nTime: " + getCurrentTime() + "\n";
```

```
    Serial.print(logEntry);
    logToSPIFFS(logEntry);

    for (int i = 0; i < 2; i++) {
      digitalWrite(BUZZER_PIN, HIGH);
      delay(100);
      digitalWrite(BUZZER_PIN, LOW);
      delay(100);
    }
  }
}

String getCurrentTime() {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    return "Unknown";
  }
  char buffer[30];
  strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M:%S", &timeinfo);
  return String(buffer);
}

void logToSPIFFS(String data) {
  File file = SPIFFS.open("/access_log.txt", FILE_APPEND);
  if (!file) {
    Serial.println("Failed to open file");
    return;
  }
  file.print(data);
  file.close();
  Serial.println("Log written to SPIFFS");
}
```