```
#include <WiFi.h>
#include <WebServer.h>
#include <FirebaseESP32.h>
#include <ArduinoJson.h>
#include <time.h>
#include <EEPROM.h>
#include <ESPping.h>

#define ONBOARD_LED 2
#define RESET_PIN 13
#define EEPROM_SIZE 512
#define WIFI_SSID_ADDR 0
#define WIFI_PASS_ADDR 64
#define FIREBASE_HOST_ADDR 128
#define FIREBASE_AUTH_ADDR 256
#define DEVICE_ID_ADDR 384
#define OWNER_UID_ADDR 416
#define NTP_SERVER1 "pool.ntp.org"
#define NTP_SERVER2 "time.google.com"
#define NTP_SERVER3 "time.nist.gov"
#define GMT_OFFSET_SEC 19800
#define DAYLIGHT_OFFSET_SEC 0
#define PRESENCE_INTERVAL 10000
#define AP_PASSWORD "esp32setup"
#define NTP_SYNC_INTERVAL 60000
#define WIFI_RETRY_INTERVAL 30000

FirebaseData fbdo;
FirebaseConfig config;
FirebaseAuth auth;
WebServer server(80);

String deviceId;
String AP_SSID;
unsigned long lastPresenceUpdate = 0;
unsigned long lastNtpSyncAttempt = 0;
unsigned long lastWiFiRetry = 0;
bool firebaseInitialized = false;

String readEEPROM(int addr, int len) {
  String data = "";
  for (int i = 0; i < len; i++) {
    char c = EEPROM.read(addr + i);
    if (c == 0) break;
```

```
    data += c;
  }
  return data;
}

void writeEEPROM(int addr, const String& data) {
  for (int i = 0; i < data.length(); i++) EEPROM.write(addr + i, data[i]);
  EEPROM.write(addr + data.length(), 0);
  EEPROM.commit();
}

void logMessage(String level, String message) {
  time_t now = time(nullptr);
  char buf[20];
  if (now > 100000) strftime(buf, sizeof(buf), "%F %T", localtime(&now));
  else strcpy(buf, "No NTP");
  Serial.printf("[%s] %s: %s\n", buf, level.c_str(), message.c_str());
}

bool hasInternet() {
  return Ping.ping("8.8.8.8", 3);
}

bool connectToWiFi() {
  String ssid = readEEPROM(WIFI_SSID_ADDR, 64);
  String pass = readEEPROM(WIFI_PASS_ADDR, 64);
  if (ssid.isEmpty()) {
    WiFi.mode(WIFI_AP);
    WiFi.softAP(AP_SSID.c_str(), AP_PASSWORD);
    logMessage("INFO", "AP Mode Started: " + AP_SSID);
    digitalWrite(ONBOARD_LED, LOW);
    return false;
  }

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid.c_str(), pass.c_str());
  logMessage("INFO", "Connecting to WiFi: " + ssid);

  int retry = 20;
  while (WiFi.status() != WL_CONNECTED && retry--) {
    digitalWrite(ONBOARD_LED, retry % 2);
    delay(500);
  }
  Serial.println();
```

```
  if (WiFi.status() == WL_CONNECTED) {
    logMessage("INFO", "Connected: " + WiFi.localIP().toString());
    digitalWrite(ONBOARD_LED, HIGH);
    return true;
  }

  WiFi.mode(WIFI_AP);
  WiFi.softAP(AP_SSID.c_str(), AP_PASSWORD);
  logMessage("INFO", "Fallback to AP Mode: " + AP_SSID);
  digitalWrite(ONBOARD_LED, LOW);
  return false;
}

void syncNTP() {
  configTime(GMT_OFFSET_SEC, DAYLIGHT_OFFSET_SEC, NTP_SERVER1, NTP_SERVER2,
NTP_SERVER3);
  for (int i = 0; i < 20; ++i) {
    if (time(nullptr) > 100000) return;
    delay(1000);
  }
  logMessage("WARN", "NTP sync failed");
}

void resetToFactory() {
  logMessage("INFO", "Factory reset...");
  for (int i = 0; i < EEPROM_SIZE; ++i) EEPROM.write(i, 0);
  EEPROM.commit();
  delay(500);
  ESP.restart();
}

void handleConfigure() {
  if (server.method() != HTTP_POST) return server.send(405);
  DynamicJsonDocument doc(1024);
  deserializeJson(doc, server.arg("plain"));

  const char* ssid = doc["ssid"];
  const char* pass = doc["password"];
  const char* host = doc["host"];
  const char* auth = doc["auth"];
  const char* userId = doc["userId"];
  const char* devId = doc["deviceId"];
```

```
  if (!ssid || !pass || !host || !auth || !userId || !devId)
    return server.send(400, "text/plain", "Missing fields");

  writeEEPROM(WIFI_SSID_ADDR, ssid);
  writeEEPROM(WIFI_PASS_ADDR, pass);
  writeEEPROM(FIREBASE_HOST_ADDR, host);
  writeEEPROM(FIREBASE_AUTH_ADDR, auth);
  writeEEPROM(OWNER_UID_ADDR, userId);
  writeEEPROM(DEVICE_ID_ADDR, devId);

  logMessage("INFO", "Saved new WiFi/Firebase config");
  server.send(200, "text/plain", "OK. Restarting...");
  delay(1000);
  ESP.restart();
}

void setup() {
  Serial.begin(115200);
  pinMode(ONBOARD_LED, OUTPUT);
  pinMode(RESET_PIN, INPUT_PULLUP);
  EEPROM.begin(EEPROM_SIZE);

  deviceId = readEEPROM(DEVICE_ID_ADDR, 32);
  if (deviceId.isEmpty()) {
    deviceId = "ESP32_" + String((uint32_t)ESP.getEfuseMac(), HEX);
    writeEEPROM(DEVICE_ID_ADDR, deviceId);
  }
  AP_SSID = "ESP32-Setup-" + deviceId;

  if (digitalRead(RESET_PIN) == LOW) resetToFactory();
  if (!connectToWiFi()) {
    server.on("/configure", HTTP_POST, handleConfigure);
    server.begin();
    return;
  }

  syncNTP();
  server.on("/configure", HTTP_POST, handleConfigure);
  server.begin();
  logMessage("INFO", "Server started");
}

void loop() {
  server.handleClient();
```

```
  if (WiFi.getMode() == WIFI_AP && millis() - lastWiFiRetry > WIFI_RETRY_INTERVAL) {
    connectToWiFi();
    lastWiFiRetry = millis();
  }
  if (time(nullptr) <= 100000 && millis() - lastNtpSyncAttempt > NTP_SYNC_INTERVAL) {
    syncNTP();
    lastNtpSyncAttempt = millis();
  }
  if (digitalRead(RESET_PIN) == LOW) {
    delay(50);
    if (digitalRead(RESET_PIN) == LOW) resetToFactory();
  }
  delay(500);
}
```