

---

# STAȚIE METEO CONECTATĂ CU JURNALIZARE DE DATE MULTI-SENZORIALĂ

---

Ștefan Diana Maria

Grupa: 343 C1

---

## INTRODUCERE: DESCRIERE GENERALĂ A PROIECTULUI ȘI OBIECTIVELE SALE

---

Proiectul propus reprezintă o stație meteorologică IoT pentru monitorizarea condițiilor ambientale, bazată pe ESP32 și utilizând senzori de temperatură, umiditate (DHT11) și presiune (BMP180). Acesta include dispozitive conectate prin Wi-Fi, care colectează și analizează datele în timp real. Datele sunt transmise într-un serviciu cloud (Firebase) pentru stocare și vizualizare, permițând analiza evoluției acestora de-a lungul timpului. Stația meteorologică creează o rețea IoT eficientă, securizată și ușor de utilizat.

Obiectivele principale ale proiectului sunt următoarele:

- Monitorizarea temperaturii și umidității: Utilizarea unui senzor DHT11 pentru a măsura temperatura și umiditatea ambientală.
- Monitorizarea presiunii: Utilizarea unui senzor BMP180 pentru a măsura presiunea.
- Transmiterea datelor: Datele colectate vor fi trimise într-o bază de date Firebase, unde vor fi stocate și pot fi accesate pentru vizualizare și analiză.
- Aplicație web: Crearea unei aplicații web care va extrage datele din Firebase și va permite vizualizarea acestora sub formă de grafice și tabele, facilitând analiza evoluției condițiilor ambientale.
- Feedback vizual cu LED RGB: Utilizarea unui LED RGB care își schimbă culoarea în funcție de temperatura măsurată, oferind un feedback vizual instantaneu despre condițiile ambientale.
- Notificări și alerte: Sistemul va trimite notificări și alerte în funcție de analiza datelor colectate de la senzori prin aprinderea unui LED RGB.
- Comunicare sigură: Asigurarea unei comunicări sigure între dispozitive și cloud prin Firebase, folosind protocoale SSL/TLS pentru protejarea datelor transmise.

Cum funcționează sistemul:

- Senzorii măsoară temperatura, umiditatea și presiunea și trimit datele către ESP32.
- ESP32 transmite aceste date către Firebase, cu un timestamp asociat.
- Aplicația web preia datele din Firebase și le vizualizează sub formă de grafice.
- Feedback-ul vizual (LED RGB) se schimbă în funcție de temperatura măsurată.
- Dacă temperatura atinge un prag critic, aplicația web trimite o alertă și LED-ul de alarmă se aprinde.

---

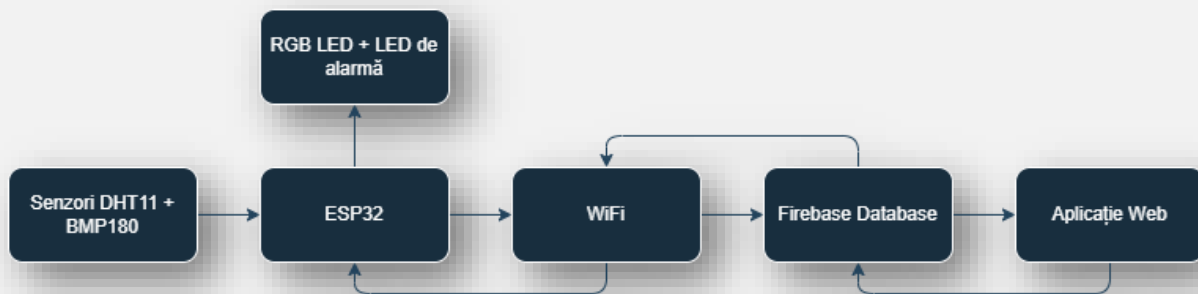
# ARHITECTURĂ: DIAGRAMA TOPOLOGIEI REȚELEI ȘI PROTOCOALELE DE COMUNICAȚIE UTILIZATE

---

## Topologia rețelei

- Senzorii (DHT11 + BMP180) sunt conectați la ESP32, care colectează datele de temperatură, umiditate și presiune.
- ESP32 se conectează la WiFi pentru a transmite datele către Firebase.
- Aplicația web:
  - Preia datele din Firebase și le vizualizează sub formă de grafice, facilitând analiza evoluției condițiilor ambientale.
  - Permite setarea unui flag pentru alarmă în Firebase, indicând o condiție critică.
- LED-uri RGB conectate la ESP32:
  - LED-ul de alarmă se va aprinde în culoarea roșu, în funcție de flagul de alarmă setat în Firebase.
  - LED-ul RGB va ajusta culoarea în funcție de temperatura măsurată, utilizând următoarele reguli:
    - Dacă temperatura este mai mare sau egală cu 30°C, LED-ul va fi roșu (pentru temperaturi ridicate).
    - Dacă temperatura este între 25°C și 30°C, LED-ul va fi galben.
    - Dacă temperatura este între 20°C și 25°C, LED-ul va fi verde.
    - Dacă temperatura este mai mică de 20°C, LED-ul va fi albastru (pentru temperaturi scăzute).

## Diagrama rețelei:



## Protocoalele de comunicație

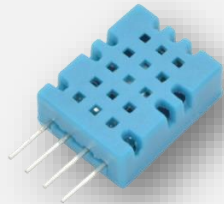
În acest proiect, protocoalele de comunicație sunt esențiale pentru transmiterea datelor între ESP32 și serviciul cloud Firebase:

- **WiFi:** Conexiunea la internet este realizată prin intermediul WiFi-ului. ESP32 utilizează un SSID și o parolă pentru a se conecta la rețeaua WiFi disponibilă. Odată conectat, ESP32 poate transmite datele colectate de la senzori către Firebase. Această conexiune este esențială pentru ca sistemul să poată funcționa într-un mediu IoT conectat la internet, permițând transferul de date în timp real.
- **HTTP/S (Firebase REST API):** Proiectul utilizează Firebase REST API pentru a trimite datele de la ESP32 către Firebase. Atunci când senzorii colectează date, acestea sunt trimise prin cereri HTTP POST către baza de date Firebase. Formatul de transmitere este JSON, care este ușor de procesat și manipulat. Protocolul HTTP/S garantează faptul că datele sunt transmise printr-o conexiune sigură, protejată de criptare.
- **Protocolul SSL/TLS:** SSL/TLS este protocolul de securitate care asigură criptarea datelor transmise între ESP32 și Firebase. Aceasta protejează datele împotriva accesului neautorizat, asigurându-se că informațiile sunt transmise într-un mod securizat, fără riscul ca acestea să fie interceptate sau modificate în tranzit. Folosirea SSL/TLS garantează că toată comunicația între dispozitivele IoT și serverele cloud este confidențială și integră.

## Componentele arhitecturii

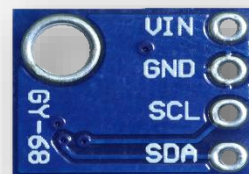
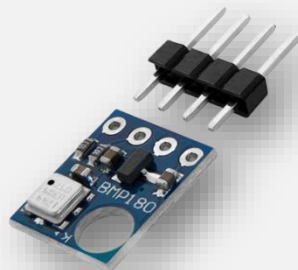
- **Senzori:**

- **DHT11** - măsoară temperatura și umiditatea. Este conectat la ESP32 și citit periodic pentru a colecta datele de mediu.



| DHT11 pins |      |
|------------|------|
| 1          | VCC  |
| 2          | DATA |
| 3          | NC   |
| 4          | GND  |

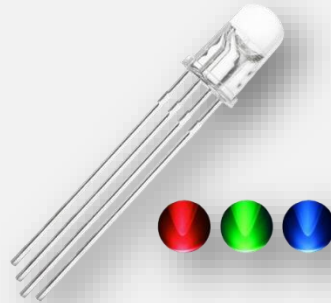
- **BMP180** - măsoară presiunea atmosferică. Este conectat la ESP32 și furnizează informații despre presiune pentru analiza condițiilor ambientale.



- **Actuatori:**

- **LED RGB** - un actuator vizual care schimbă culoarea în funcție de temperatura citită de la senzorul DHT11. Acesta furnizează un feedback vizual imediat al condițiilor de mediu, ajutând utilizatorul să înțeleagă rapid schimbările de temperatură.

- **LED de alarmă** - un actuator care se aprinde atunci când temperatura, umiditatea sau presiunea ating o valoare critică. Acesta este controlat printr-un flag în Firebase setat de aplicația web.



- **Aplicația de vizualizare a datelor:**

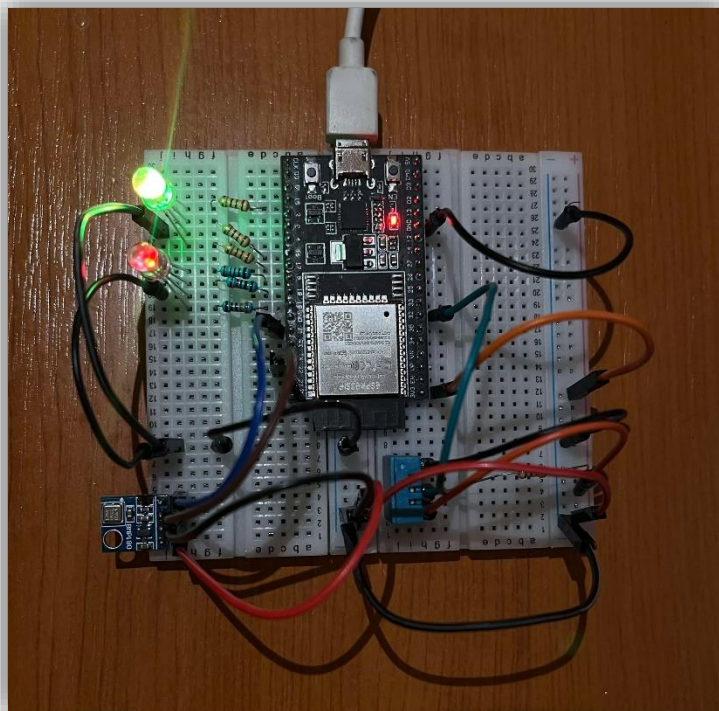
- Aplicația web se conectează la Firebase pentru a prelua datele de la senzori (temperatură, umiditate, presiune) și le vizualizează sub formă de grafice, oferind utilizatorului o interfață prietenoasă pentru monitorizarea condițiilor ambientale.
- Aplicația web permite utilizatorului să seteze un flag pentru alarmă în Firebase, care controlează aprinderea LED-ului de alarmă.

---

## IMPLEMENTARE: PAȘII DE CONFIGURARE A HARDWARE-ULUI, SOFTWARE-ULUI, ȘI SISTEMULUI DE ALERTARE ȘI NOTIFICARE

---

### Configurarea hardware-ului:



- **ESP32:** Este microcontrolerul folosit pentru proiect. Acesta oferă suport pentru Wi-Fi și este responsabil pentru colectarea datelor de la senzori și trimiterea acestora către Firebase. În plus, ESP32 controlează LED-ul RGB pentru feedback vizual și gestionează comunicarea cu senzorul BMP180 și cu senzorul DHT11.



- **Senzori:**

- DHT11:

Senzorul DHT11 măsoară temperatura și umiditatea. Conectarea acestui senzor la ESP32 se face astfel:

- Pinul de semnal (data) al senzorului DHT11 este conectat la GPIO 32 al ESP32.
- Pinul VCC este conectat la 3.3V pe ESP32 pentru alimentare.
- Pinul GND este conectat la GND pe ESP32.

Senzorul DHT11 comunică datele, iar ESP32 le preia și le trimite mai departe la Firebase și pentru a fi utilizate pentru feedback vizual.

- BMP180: Măsurarea presiunii atmosferice.

Senzorul BMP180 măsoară presiunea atmosferică și poate calcula altitudinea. Acesta este conectat la ESP32:

- SDA (Serial Data) al senzorului BMP180 este conectat la GPIO 21 al ESP32.
- SCL (Serial Clock) al senzorului BMP180 este conectat la GPIO 22 al ESP32.
- Pinul VCC este conectat la 3.3V, iar pinul GND la GND.

Senzorul BMP180 transmite datele către ESP32, iar ESP32 preia aceste date și le folosește pentru a calcula presiunea atmosferică și altitudinea, trimițându-le la Firebase.

- **LED RGB (Semnalizare alarmă):**

Acest LED RGB se va aprinde roșu atunci când un flag de eroare este detectat, indicând o alarmă sau o condiție anormală. În rest, acest LED va rămâne oprit

- Pinul Roșu al LED-ului RGB este conectat la GPIO 19.
- Pinul Verde este conectat la GPIO 18.
- Pinul Albastru este conectat la GPIO 5.

- **LED RGB (Feedback pentru temperatură):**

Ledul va schimba culoarea în funcție de temperatura măsurată de senzorul DHT11:

- Pinul Roșu este conectat la GPIO 17.
- Pinul Verde este conectat la GPIO 16.
- Pinul Albastru este conectat la GPIO 15.

Ledul afișează o culoare diferită în funcție de temperatura măsurată:

- **Roșu:** pentru temperaturi de 30°C sau mai mari.
- **Galben:** pentru temperaturi între 25°C și 30°C.
- **Verde:** pentru temperaturi între 20°C și 25°C.
- **Albastru:** pentru temperaturi sub 20°C.

Fiecare culoare (roșu, verde, albastru) este controlată prin PWM.

```
pins used

// Define DHT pin and type
#define DHTPIN 32 // GPIO pin where the DHT11 is connected
#define DHTTYPE DHT11 // Define the type of DHT sensor

// Define pins used for BMP sensor
#define BMP_SDA 21 // GPIO pin for BMP sensor SDA
#define BMP_SCL 22 // GPIO pin for BMP sensor SCL

// Define RGB LED pins
#define BLUE_PIN 5 // GPIO pin for Red
#define GREEN_PIN 18 // GPIO pin for Green
#define RED_PIN 19 // GPIO pin for Blue

// Define pins for the temperature RGB LED
#define TEMP_BLUE_PIN 15 // GPIO pin for Blue temperature LED
#define TEMP_GREEN_PIN 16 // GPIO pin for Green temperature LED
#define TEMP_RED_PIN 17 // GPIO pin for Red temperature LED
```

## Configurarea software-ului - firmware.ino:

### Librării utilizate:

- **DHT.h** și **DHT\_U.h**: Pentru citirea datelor de la senzorul DHT11.
- **WiFi.h**: Pentru conectarea ESP32 la rețeaua WiFi.
- **FirestoreESPClient.h**: Pentru interacțiunea cu Firebase.
- **Adafruit\_BMP085\_U.h**: Pentru citirea presiunii atmosferice de la senzorul BMP180.

### Conexiune WiFi:

- ESP32 se conectează la o rețea WiFi specificată prin **SSID** și **parolă**.

### Configurare Firebase:

- Accesul la datele din Firebase este realizat prin configurarea unui **API Key** și **URL-ul bazei de date**.

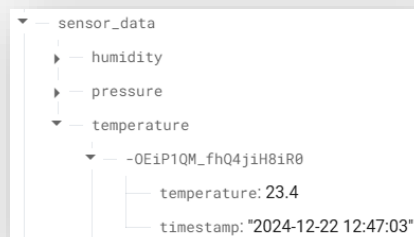
### Fluxul codului

#### 1. Colectarea datelor de la senzori:

- Senzorul DHT11 citește temperatura și umiditatea.
- Senzorul BMP180 citește presiunea atmosferică.

#### 2. Transmiterea datelor:

- Datele colectate sunt trimise la Firebase sub forma unui obiect JSON, care include și timestamp-ul.



### 3. Schimbarea culorii LED-ului RGB:

- În funcție de valoarea temperaturii, LED-ul RGB își schimbă culoarea.

### 4. Sincronizarea timpului:

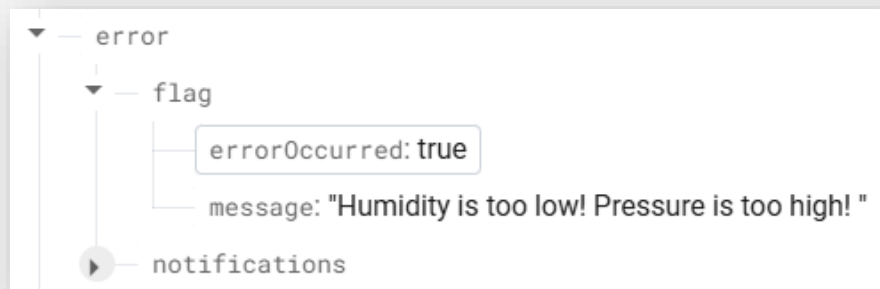
- Ora este sincronizată cu un server NTP pentru a asigura timestamp-uri precise pentru datele stocate în Firebase.

### 5. Gestionarea erorilor:

- Codul verifică erorile senzorilor și afișează mesaje în monitorul serial dacă citirile sunt invalide.

### 6. Funcționalități

- **getCurrentTime()**: Obține timpul curent de la serverul NTP și îl formatează ca timestamp.
- **checkSensorData()**: Verifică validitatea citirilor senzorilor.
- **pushToFirebase()**: Trimite datele (temperatură, umiditate, presiune) la Firebase în format JSON, incluzând un timestamp.
- **setColor()**: Controlează culoarea LED-ului RGB.
- **getTemperatureColor()**: Determină valorile RGB pe baza temperaturii măsurate.
- **checkErrorFlag()**: Verifică flagul de eroare din Firebase și afișează mesajul asociat, dacă este activat.



## Sistem de alertă și notificare:

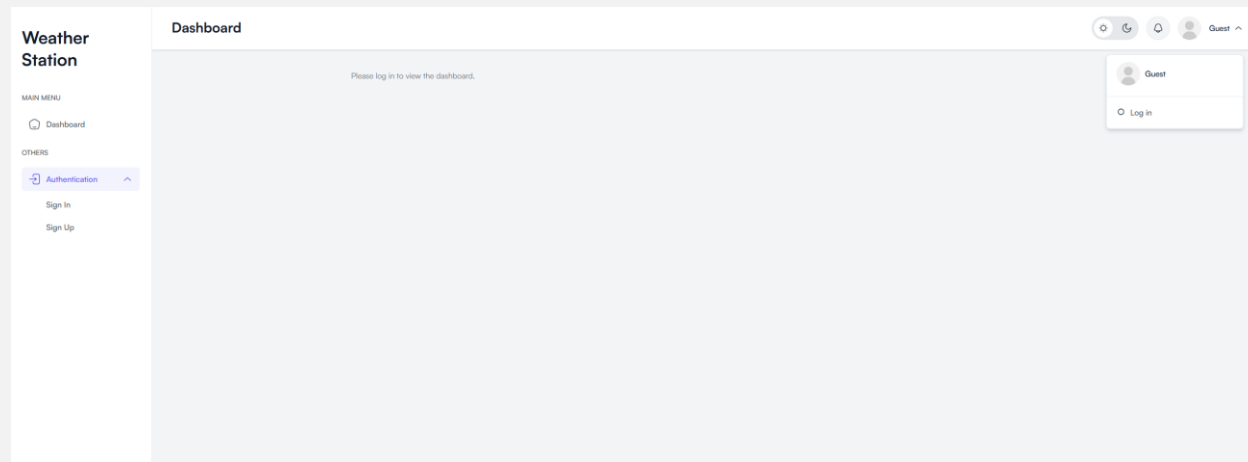
- Un sistem de alertă este configurat astfel încât, în cazul unui senzor care depășește un anumit prag, o notificare este trimisă utilizatorului prin intermediul culorii LED-ului RGB.
- Datele sunt trimise către Firebase cu un timp de trimitere verificat periodic pentru a evita o suprasolicitare a bazei de date.

# VIZUALIZARE ȘI PROCESARE DE DATE: EXPLICAREA METODEI DE PROCESARE ȘI AFIȘARE A DATELOR SENZORILOR ÎNTR-O INTERFAȚĂ INTUITIVĂ

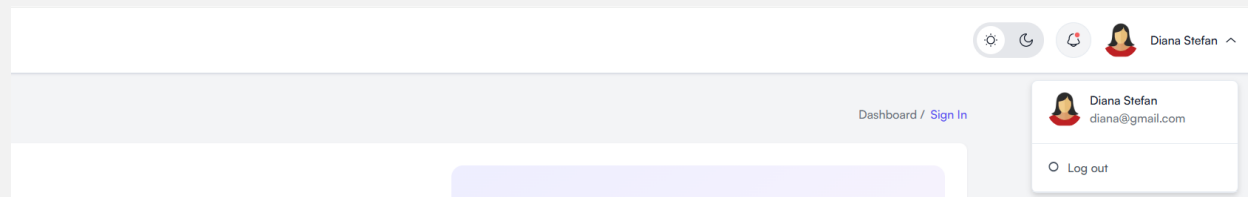
Aplicația web creată pentru vizualizarea datelor colectate de la senzorii DHT11 și BMP180 este concepută pentru a oferi o interfață intuitivă și ușor de utilizat. Aceasta procesează datele senzorilor, trimise către Firebase și le afișează într-un dashboard interactiv, cu o serie de funcționalități pentru a monitoriza condițiile ambientale și pentru a gestiona alertele și notificările.

## 1. Autentificarea utilizatorilor:

- La accesarea aplicației, utilizatorul este întâmpinat de o **secțiune de autentificare**. Dacă nu este autentificat, aplicația afișează un mesaj clar: *"Please log in to view the dashboard"*.



- Pentru a se autentifica, utilizatorul poate accesa opțiunile **Sign In** (pentru utilizatori existenți) sau **Sign Up** (pentru crearea unui cont nou). De asemenea, poate alege să se autentifice rapid accesând iconița de profil).
- Autentificarea se realizează prin intermediul Firebase, care gestionează siguranța și protecția datelor utilizatorilor.



Email

Password


Sign In

[Don't have an account? Sign Up](#)

Sign in to your account

## Welcome Back!

Please sign in to your account by completing the necessary fields below



Name

Email

Password

Gender

Select your gender

▼


Sign Up

[Already have an account? Sign In](#)

Sign up to your account

## Welcome!

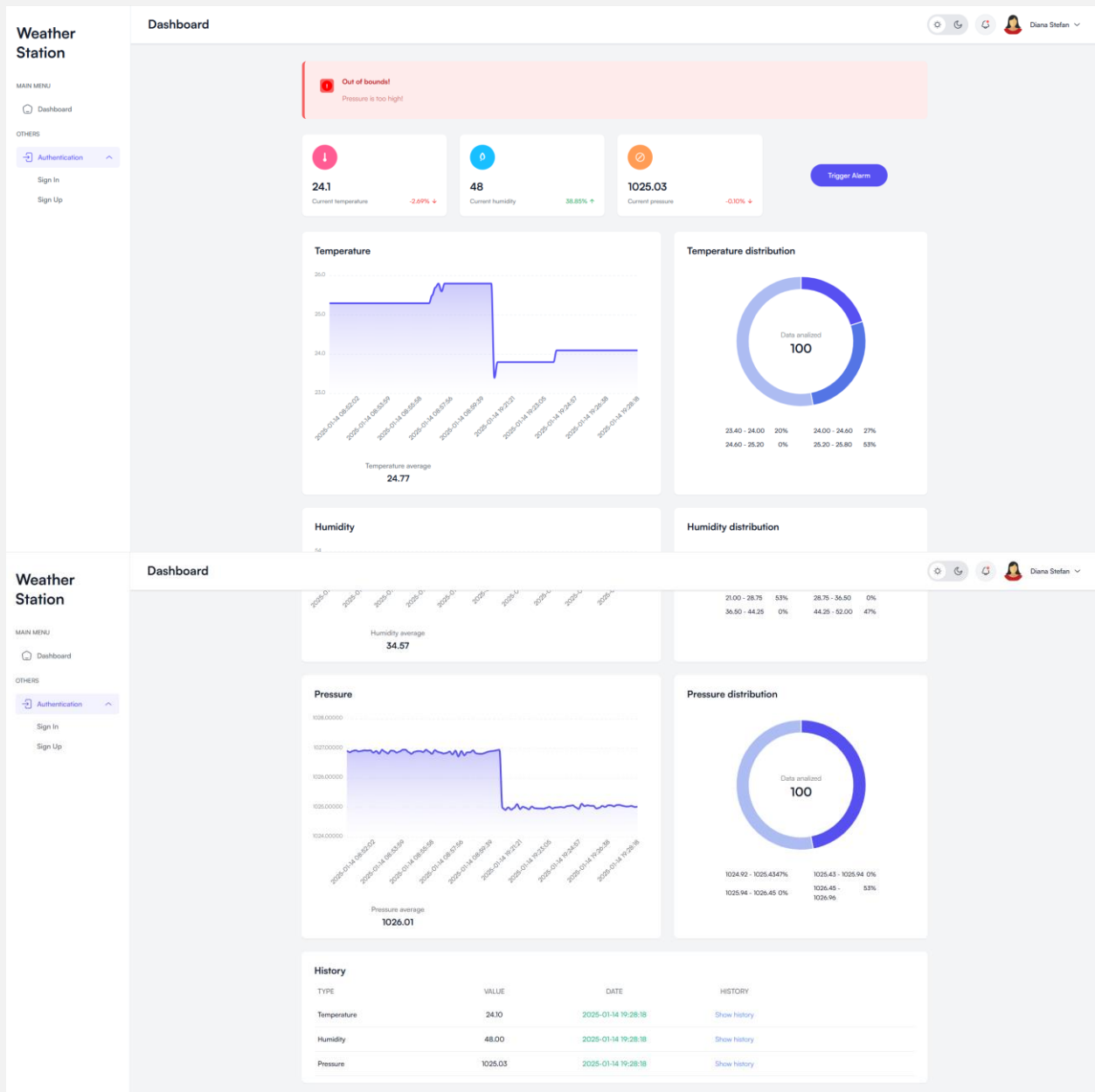
Please sign up by completing the necessary fields below



2. **Dashboard-ul principal:** După autentificare, utilizatorul poate accesa **dashboard-ul** principal, care oferă o vizualizare completă a datelor senzorilor și a alertelor:

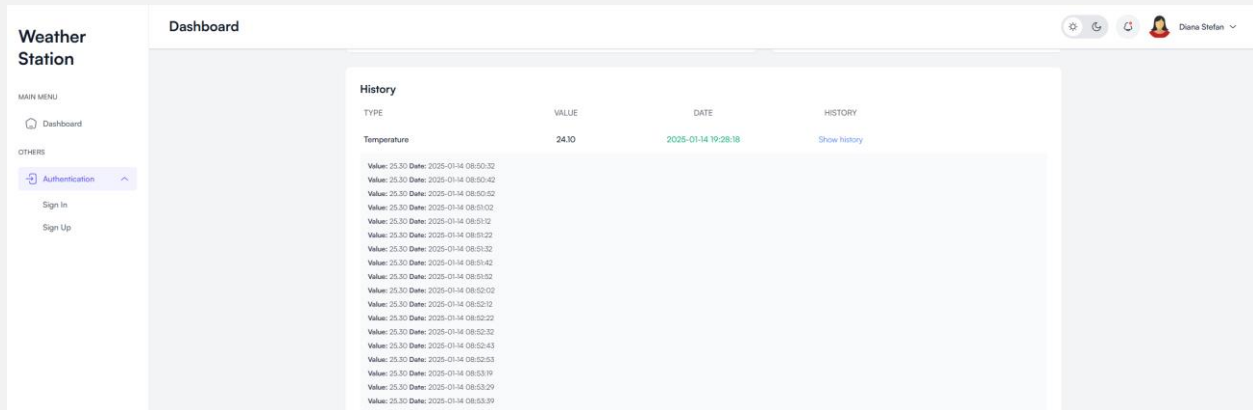
- **Alarmerle:** Dacă valorile măsurate de senzorii de temperatură, umiditate sau presiune depășesc pragurile stabilite, aplicația va declanșa o alarmă vizuală. Aceste alarme sunt afișate în dashboard pentru a atrage atenția asupra valorilor neobișnuite.
- **Chenare cu valorile curente:** În dashboard, utilizatorul poate vedea valorile curente pentru temperatura, umiditate și presiune, alături de un procentaj ce indică modificările față de media istoricului de date. Aceasta ajută la înțelegerea rapidă a tendințelor și schimbărilor ambientale.

- **Grafice:** Sunt afișate grafice pentru ultimele 100 de înregistrări ale datelor de la senzori, permițând utilizatorului să observe evoluția acestora pe termen scurt. Graficele includ și distribuția valorilor pentru fiecare senzor, pentru a ajuta utilizatorul să vizualizeze clar variațiile în date.





- **Tabel cu intrările efective:** Un tabel detaliat prezintă intrările istorice pentru temperatură, umiditate și presiune, împreună cu timestamp-urile asociate fiecărei măsurători. Acesta este folosit pentru a accesa rapid datele dintr-o perioadă anume și pentru a analiza tendințele în timp.

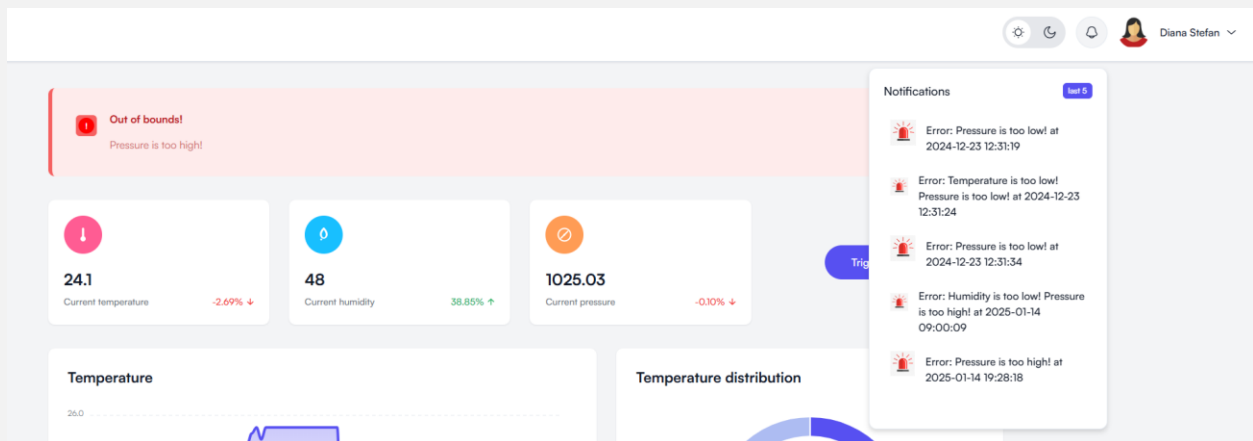


The screenshot shows the 'Weather Station' dashboard. On the left is a sidebar with 'MAIN MENU' (Dashboard) and 'OTHERS' (Authentication, Sign In, Sign Up). The main area is titled 'Dashboard' and contains a 'History' table. The table has columns: TYPE, VALUE, DATE, and HISTORY. The first row shows 'Temperature' with a value of '24.30' and a date of '2025-01-14 19:28:18'. Below this is a list of 20 rows, each with a 'Value' and a 'Date'.

| TYPE         | VALUE                     | DATE                | HISTORY      |
|--------------|---------------------------|---------------------|--------------|
| Temperature  | 24.30                     | 2025-01-14 19:28:18 | Show history |
| Value: 25.30 | Date: 2025-01-14 08:50:32 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:50:42 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:50:52 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:51:02 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:51:12 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:51:22 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:51:32 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:51:42 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:51:52 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:52:02 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:52:12 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:52:22 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:52:32 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:52:43 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:52:53 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:53:09 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:53:29 |                     |              |
| Value: 25.30 | Date: 2025-01-14 08:53:39 |                     |              |

### 3. Notificări și alerte:

- În colțul superior al dashboard-ului, există un **buton de notificări** care deschide un dropdown cu alertele salvate în Firebase. Aceste alerte sunt generate pe baza pragurilor de temperatură, umiditate și presiune setate de utilizator. Dacă valorile senzorilor depășesc aceste praguri, aplicația va adăuga o notificare în dropdown, iar utilizatorul va fi informat despre evenimentele importante.



The screenshot shows the 'Weather Station' dashboard with a notification dropdown open. The dashboard displays three sensor cards: 'Current temperature' (24.1, -2.69% ↓), 'Current humidity' (48, 38.85% ↑), and 'Current pressure' (1025.03, -0.10% ↓). Below these are two charts: 'Temperature' and 'Temperature distribution'. The notification dropdown shows five alerts: 'Error: Pressure is too low! at 2024-12-23 12:31:19', 'Error: Temperature is too low! Pressure is too low! at 2024-12-23 12:31:24', 'Error: Pressure is too low! at 2024-12-23 12:31:34', 'Error: Humidity is too low! Pressure is too high! at 2025-01-14 09:00:09', and 'Error: Pressure is too high! at 2025-01-14 19:28:18'.

## Implementare (React + Next.js):

### 7. Componenta principală:

- Dashboard-ul este centrul aplicației și include informații esențiale despre condițiile meteo, prezentate prin grafice și tabele interactive.
- **ApexCharts** este folosit pentru a crea grafice liniare care urmăresc evoluția în timp a parametrilor meteo (temperatură, umiditate, presiune). Aceste grafice sunt interactive și responsive, permițând utilizatorului să vizualizeze datele într-un mod dinamic. **ReactApexChart** este integrat pentru a gestiona graficul în componenta React.
- Datele pentru grafice sunt extrase din **Firebase** și sunt actualizate în timp real, ceea ce permite o vizualizare continuă a schimbărilor din parametrii meteo.
- De asemenea, în dashboard sunt incluse **tabele** care organizează datele într-un format ușor de citit. Tabelele sunt actualizate automat pe măsură ce noi date sunt disponibile, iar utilizatorul poate accesa informații detaliate despre fiecare măsurătoare.

### 8. Interfața de utilizator:

- Folosirea **PrimeReact** pentru componente vizuale, cum ar fi carduri, butoane și tabele.
- Design responsiv folosind **Tailwind CSS** pentru a asigura o experiență bună pe toate dispozitivele.

### 2. Configurarea Firebase

- Aplicația se bazează pe Firebase pentru gestionarea autentificării și stocarea datelor. Configurarea Firebase include inițializarea aplicației Firebase cu cheile API și setările de configurare necesare pentru conectarea la Realtime Database și Firestore.
- **Fișierul firebaseConfig.js:**
  - În acest fișier sunt incluse cheile și setările pentru autentificarea Firebase, Realtime Database și Firestore.

### 3. Autentificarea utilizatorilor cu Firebase Authentication

Pentru a permite autentificarea utilizatorilor, aplicația utilizează **Firebase Authentication**. S-au implementat următoarele funcționalități:

- **Pagina de autentificare:** Permite utilizatorilor să se logheze folosind un email și o parolă. Datele sunt validate, iar la succes utilizatorul este redirecționat către dashboard.

---

## SECURITATE: MĂSURI DE SECURITATE IMPLEMENTATE

---

Pentru a asigura protecția datelor și securitatea aplicației, au fost implementate mai multe măsuri de securitate esențiale. Aceste măsuri vizează criptarea datelor în tranzit, autentificarea utilizatorilor și protejarea accesului la Firebase. Detaliile pentru fiecare măsură sunt prezentate mai jos:

### Criptare (SSL/TLS)

Toate comunicațiile între ESP32 și Firebase sunt securizate automat de Firebase utilizând protocolul SSL/TLS (Secure Sockets Layer/Transport Layer Security). Aceasta este o măsură de securitate standard care garantează criptarea datelor în timpul transmiterii, prevenind interceptarea și accesul neautorizat.

- **SSL/TLS în transmisia HTTP:** Atunci când ESP32 trimite date către Firebase prin HTTP, cererile sunt automat criptate prin SSL/TLS. Astfel, informațiile trimise, precum temperatura, umiditatea și presiunea, sunt protejate și nu pot fi citite sau modificate de terți în timpul transmiterii.
- **Confidențialitate și integritate:** Datorită criptării SSL/TLS, datele rămân confidențiale și integritatea lor este asigurată, protejându-le astfel de orice modificări sau acces neautorizat în timpul transmiterii între ESP32 și Firebase.
- **Prevenirea atacurilor MitM:** Criptarea SSL/TLS protejează comunicațiile împotriva atacurilor de tip Man-in-the-Middle (MitM), unde un atacator ar putea intercepta și modifica datele transmise.

## Autentificare și Acces Controlat

Pentru a proteja accesul la baza de date Firebase și a preveni accesul neautorizat, se utilizează un sistem de autentificare bazat pe un API key valid. Acesta controlează permisiunile de acces la resursele Firebase.

- **API Key pentru autentificare:** Fiecare aplicație care se conectează la Firebase utilizează un API key, generat automat de Firebase în momentul creării proiectului. Acest API key acționează ca un identificator unic pentru aplicația respectivă.
- **Autentificare la Firebase:** API key-ul este inclus în cererile HTTP trimise de ESP32 către Firebase. Acesta garantează că doar aplicațiile autorizate pot accesa datele. Firebase verifică acest API key înainte de a permite accesul la baza de date.
- **Permițiuni și securitate:** Firebase permite configurarea unor reguli de securitate la nivelul bazei de date, prin care se limitează accesul la resurse doar la cererile provenite de la ESP32 și care sunt autentificate corect. Aceste reguli previn accesul neautorizat la datele stocate și permit doar operațiuni de citire și scriere sigure.

## Autentificarea utilizatorilor pentru vizualizarea graficelor

Pentru a asigura accesul doar utilizatorilor autorizați la vizualizarea datelor și a graficelor, autentificarea utilizatorilor este gestionată direct de Firebase. Astfel, doar utilizatorii care se autentifică corect pot vizualiza graficele și informațiile sensibile stocate în aplicație.

- **Autentificare prin Firebase:** Firebase se ocupă de procesul de autentificare a utilizatorilor, asigurându-se că numai utilizatorii autentificați pot accesa secțiunile protejate ale aplicației, inclusiv vizualizarea datelor în grafice. Acest sistem de autentificare include opțiuni precum autentificarea prin email și parolă sau prin metode de autentificare externe (Google, Facebook etc.).
- **Acces restricționat la grafice:** Graficile cu datele senzorilor sunt accesibile doar utilizatorilor autentificați. Aceasta asigură că informațiile sensibile nu sunt accesibile persoanelor neautorizate.

---

## PROVOCĂRI ȘI SOLUȚII: PROBLEME ÎNTÂMPINATE ȘI SOLUȚIILE APLICATE

---

### Probleme întâmpinate:

- Conectivitate WiFi instabilă:
  - Pierderea conexiunii WiFi a dus la întreruperea transmisiei datelor către Firebase.
- Citirea incorectă a datelor de la senzori (DHT11, BMP180):
  - Valorile citite de la senzori erau adesea invalide, ceea ce afecta acuratețea datelor trimise.
- Nestăpânirea tehnologiilor web:
  - Dificultăți în utilizarea unor tehnologii web avansate pentru vizualizarea și procesarea datelor senzorilor.

### Soluții aplicate:

- Conectivitate WiFi:
  - Utilizarea unui cod repetitiv pentru a reconecta ESP32 la WiFi în cazul pierderii conexiunii, asigurându-se astfel o conectivitate constantă.
- Citirea datelor senzorilor:
  - Implementarea unor verificări suplimentare pentru validarea datelor de la senzori înainte de trimiterea acestora către Firebase.
  - În cazul în care datele sunt invalide (de exemplu, temperatura sau umiditatea citite sunt NaN), sistemul va afișa un mesaj de eroare și va evita trimiterea datelor incorecte.
- Nestăpânirea tehnologiilor web:
  - Documentarea pe internet și urmarea unor tutoriale.