

SENZOR MONITORIZARE BEBELUSI

ILIE ANDREI-LEONARD 325CB – ADIV

Tema proiectului este realizarea unui sistem de monitorizare a bebelusilor in timp real. In brieful initial am mentionat si realizarea unei aplicatii de Android, insa in timpul realizarii proiectului, mi s-a stricat modulul de bluetooth, asa ca a trebuit sa schimb putin partea de conectivitate, si am migrat spre o platforma web (cross-platform- compatibila si cu telefoane si cu desktop/laptop).

Proiectul este compus din 3 parti:

- Partea hardware, unde am folosit o placut de Arduino Uno si un modul de microfon.

Modulul de microfon a avut 4 pini, insa am folosit doar 3 dintre ei (unul fiind digital). Astfel:

- Plusul de pe microfon l-am conectat la 5V.
- GND la GND
- Si analogicul, prin care vom transmite la server datele din microfon l-am conectat pe pinul A0.

Am scris un cod simplu de Arduino pentru a programa placuta sa afiseze decibelii primiti de la microfon. (am folosit o formula de pe Google de transformare a informatiei primite in decibeli).

Tot proiectul poate fi regasit atat pe github - <https://github.com/sKzRO/Senzor-Monitorizare-Bebelusi>, dar voi anexa bucati de cod si in aceasta documentatie.

Pentru Arduino codul folosit a fost urmatorul:

```
const int analogInPin = A0; // Inputul analogic pe care am pus microfonul
```

```
int sensorValue = 0; // Variabila unde vom citi valoarea din senzor
```

```
int prevsensorValue = 0;
```

```
void setup() {
```

```
    // Citim la 9600 biti
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```

// Intr-un loop infinit citim valoarea din microfon
sensorValue = analogRead(analogInPin);

int sensorValueInDB = (sensorValue+83.2073) / 11.003; // Formula de pe Google

// Printam valoarea
Serial.print(sensorValueInDB);

// Facem delay ca sa nu spamam
delay(100);
}

```

Practic avem set-up-ul unde facem setarea initiala, apoi intram in loop unde citim informatia din microfon, o transformam in decibeli.

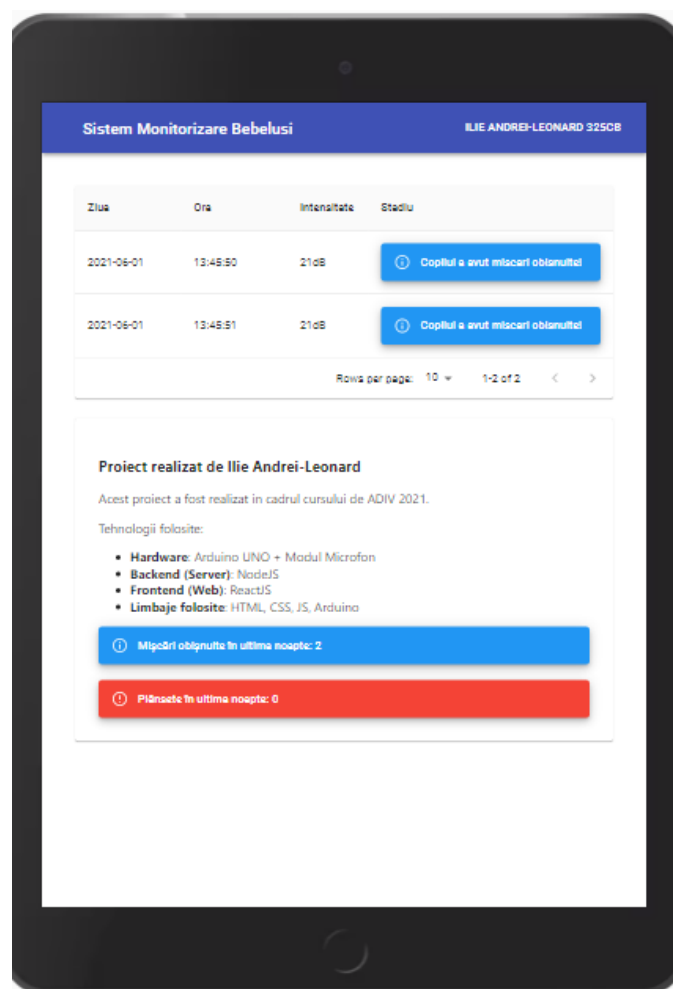
- Partea de server este realizata in Javascript (NodeJS), si folosind biblioteca SerialPort, am putut prelucra informatiile primite dintr-un port (dupa ce am incarcat fisierul de Arduino pe placuta).

Fisierul app.js din Backend este singurul fisier de backend, in el deschidem portul, si ascultam de el. In momentul in care serverul primeste un semn din partea portului, verifica daca valoarea in decibeli este mai mare de 20 (ca sa evitam spamul la server), iar daca este mai mare, folosind Socketi (o metoda prin care comunicam in timp real intre client browser si server) trimitem informatia catre browser (unde va fi prelucrata de frontend). Pentru proiect nu am folosit o baza de date in care sa tin informatiile primite, inasa tin informatiile intr-un storage (state) in browser.

- Partea de browser (frontend) este realizata tot in Javascript (folosind ReactJS). La aceasta parte s-a muncit cel mai mult, pentru ca pe langa aspectele de implementare, a trebui sa tin cont si de componenta de responsive (dupa cum am zis proiectul este cross-platform).

In frontend, avem folderul src, unde avem componentele principale ale aplicatiei. App.js contine componentele principale (Header si Aplicatie – aplicatia in sine), iar in Aplicatie este partea functionala.

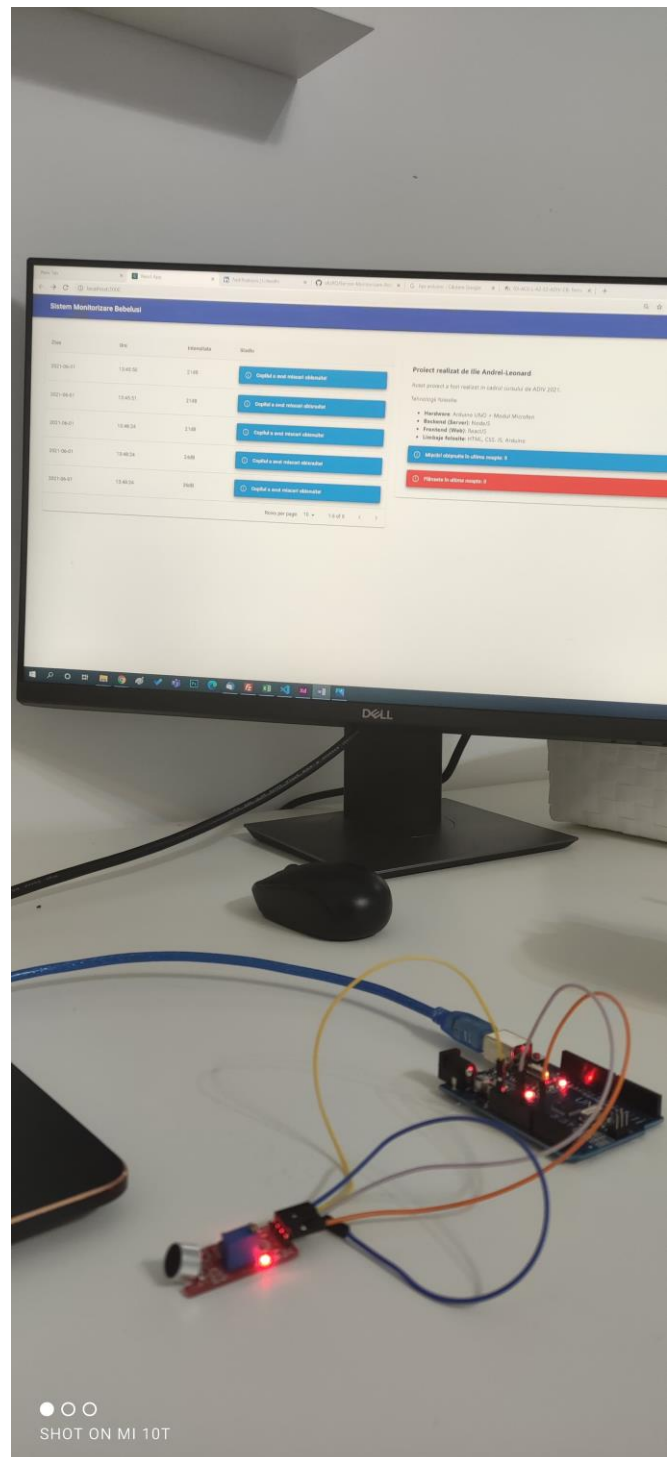
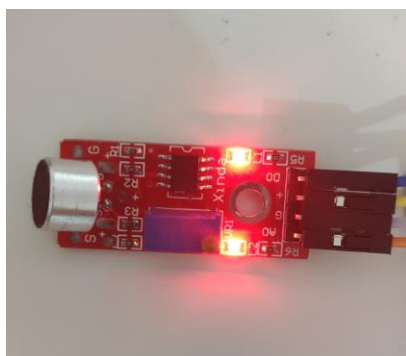
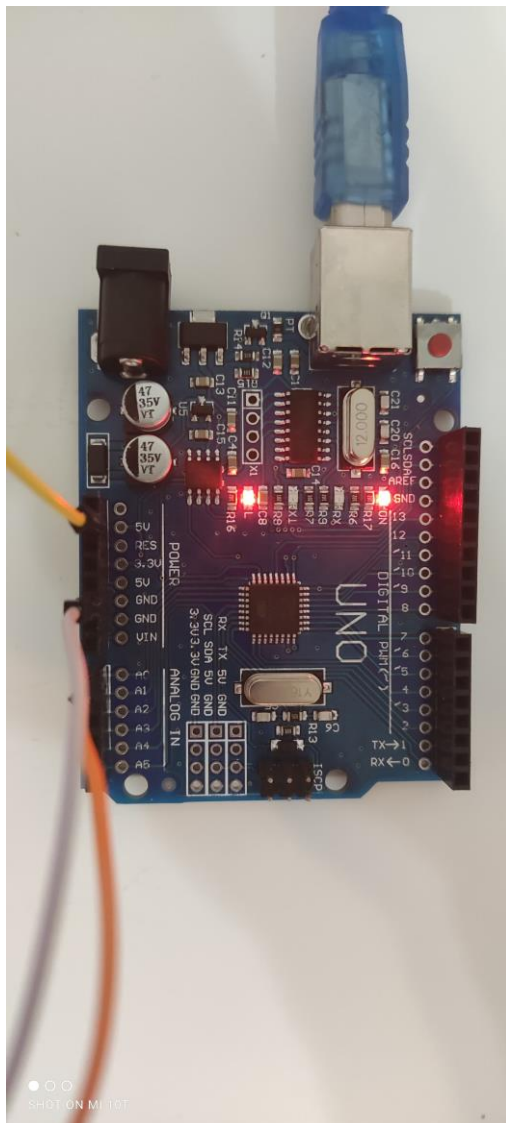
Folosind useEffect din React (un callback care ne permite ca de fiecare data cand este accesat sa modifice interfata), conectam socket-ul la server, si de fiecare data cand primim un ping de la server (care primeste la randul lui informatie din microfon) testeaza decibelii, si in functie de asta adauga in tabel datele. Am declarat doua tipuri de date ale bebelusilor care pot fi monitorizate (miscari obisnuite si miscari alarmante – plansete). Anexez mai jos o poza cu interfata pe care o vede utilizatorul (mobile + desktop) si voi explica pe ea toate elementele.



Putem observa in platforma 2 componente principale, la care putem aduce cu succes alte componente (aplicatia fiind scalabila).

Prima componenta este un tabel cu ziua, data, intensitatea sunetului si tpiul de notificare pe care o primeste parintele, iar a doua este un centralizator, prin care parintele vede cate notificari a primit noaptea trecuta de la bebelus.

Tabelul este dinamic, se pot seta cate date sa fie pe pagina, este integrata paginatie, ca sa nu scrolezi foarte mult.



Bibliografie:

- <https://nodejs.org/en/docs/> - documentatie backend
- <https://reactjs.org/docs/getting-started.html> - documentatie frontend
- <http://google.com/> - formula decibeli
- <https://docs.arduino.cc/> - documentatie arduino
- <https://material-ui.com/> - componente de design website

Concluzie:

In concluzie, folosind un microfon fin, si proiectul dezvoltat de mine, putem monitoriza cu succes somnul unui bebelus, putem primi notificari, daca avem o casa mai mare si bebelusul doarme in alta camera, asta cu un buget foarte redus.

COD PROIECT: <https://github.com/sKzRO/Senzor-Monitorizare-Bebelusi>