

Laborator 6

Web App

Obiective

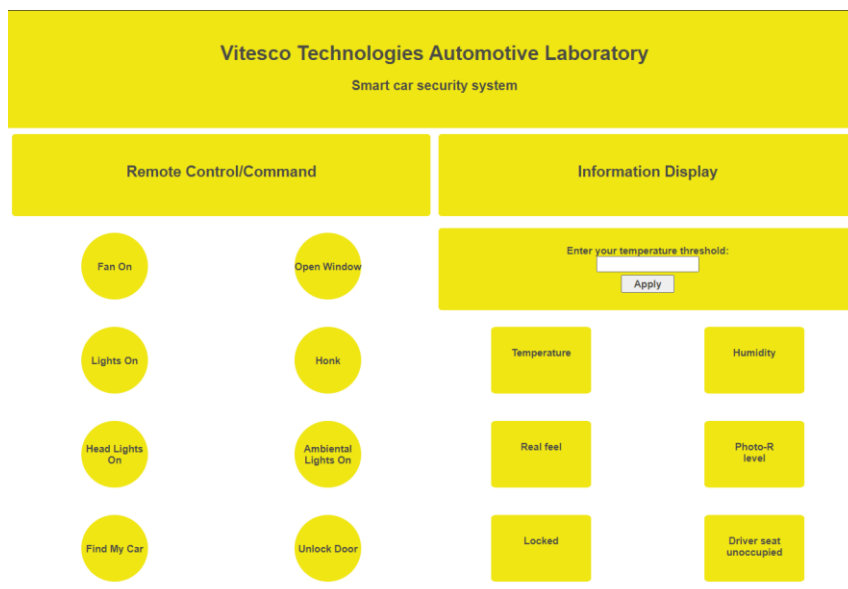
- Înțelegerea structurii și funcționării aplicației web

Cuprins

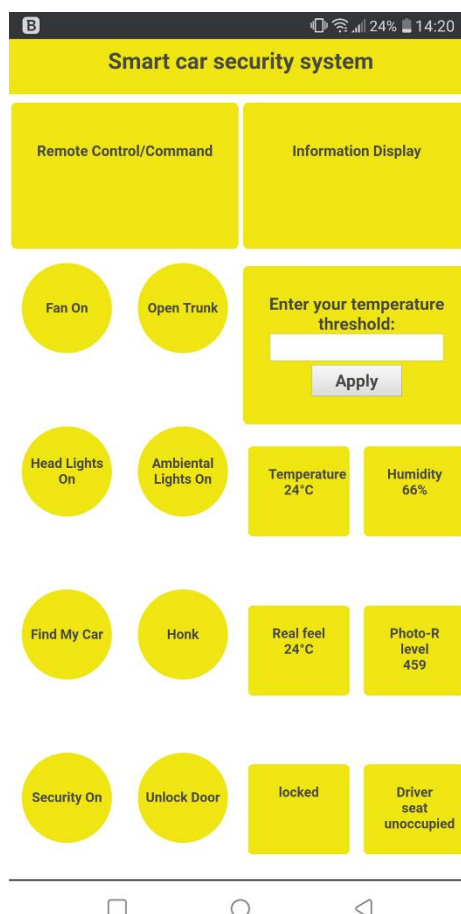
Obiective	1
Cuprins	1
Structura	2
Funcționalitate	4

Structura

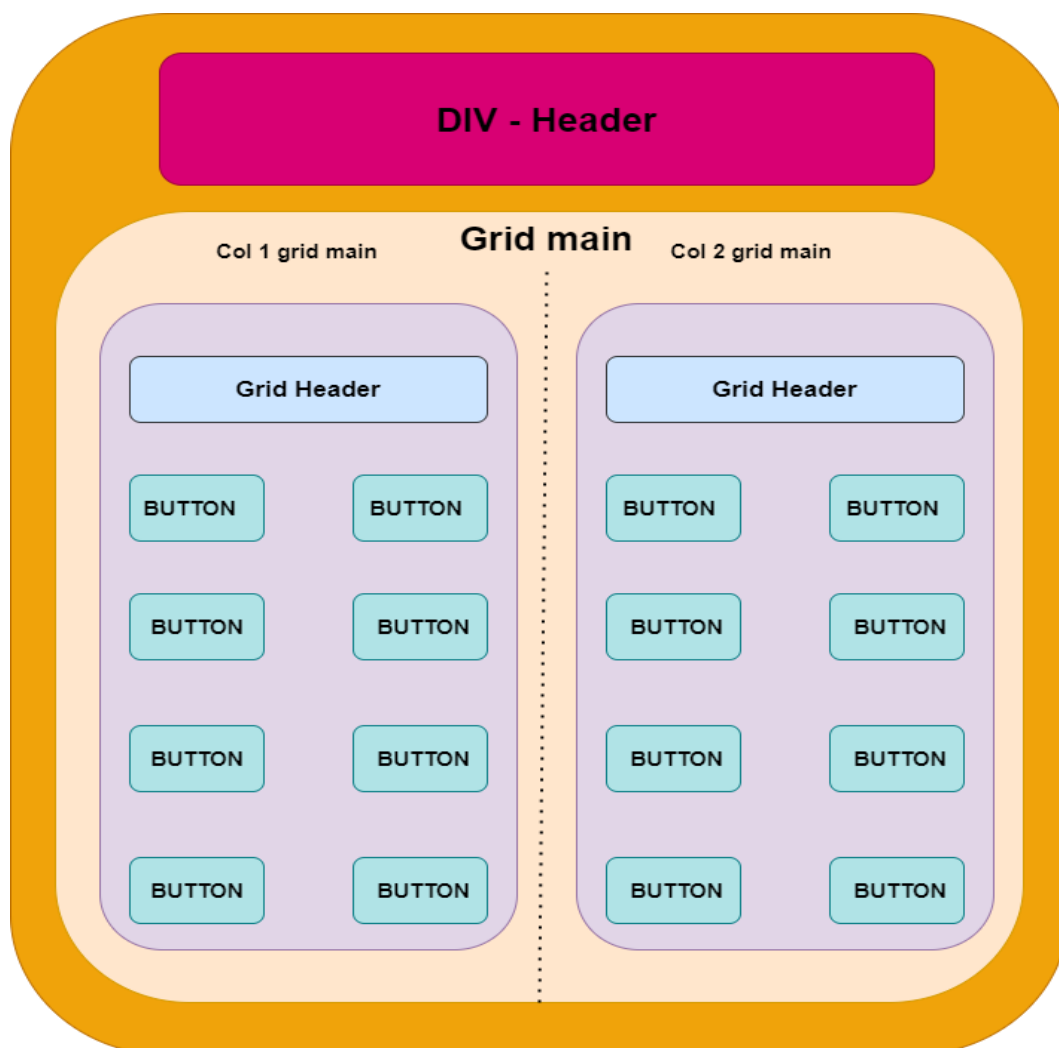
Aplicația văzută pe desktop:



Aplicația văzută pe telefon:



Aplicația are următoarea structură:



Partea de sus, header-ul, este doar un simplu div ce conține două elemente de tipul h1,h2.

Cea mai mare parte a aplicației este formată din grid-uri, așadar:

- Tot container-ul mare ce se întinde pe toată lățimea aplicației este un div afișat ca grid cu 2 coloane si un singur rând ce are clasa ***class= "grid-container-main"***
- Cele 2 coloane, la rândul lor, reprezintă tot câte un grid cu clasele:
 - ***class= "grid-container-left"*** este un grid cu 2 coloane și 5 rânduri, fiecare element din acest grid, exceptând primul element ce are rol de header al container-ului ce se întinde pe tot rândul 1, este o succesiune de forma `div>form>button`, adică un buton ce are ca parinte un element `form` care la rândul lui are ca părinte un element de tipul `div`. Acest container are ca rol controlul componentelor (senzori/motoare/leduri) conectate la microcontroler

```
.grid-container-left {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: 20% 1fr 1fr 1fr 1fr;
}
```

```
<div class="grid-item">
  <div class="form-item">
    <form id="buttonFormFan" method="post">
      <button class="buttonSubmit" type="submit" id="buttonSubmitFan" name="button" value="Fan On">Fan On</button>
    </form>
  </div>
</div>
```

- ***class="grid-container-right"*** este de asemenea un grid cu 5 randuri și 2 coloane, dar fiecare element aici este format dintr-un paragraf ce are ca părinte un div, iar pe ultimul rând, nu este afișat nimic. Acest container are ca rol afișarea informațiilor primite de la microcontroller

Funcționalitate

Funcționalitatea este realizată cu ajutorul unui script, prin care se actualizează în timp real informațiile afișate pe aplicația web.

Transmiterea de comenzi de la aplicația web la server (esp8266) se realizează prin intermediul unei funcții ce se apelează automat la detectarea unui eveniment (submit), pentru fiecare buton este implementată o astfel de funcție. În urma apelării funcției se vor actualiza valorile butoanelor. Valorile atribuite butoanelor se vor transmite către server (eveniment de tip POST) sub forma “name=value”.

De exemplu:

```
document.getElementById('buttonFormFan').addEventListener('submit', buttonEventFan);
if (localStorage.getItem("valueFan") == null) localStorage.setItem("valueFan", "Fan On");
document.getElementById('buttonSubmitFan').innerHTML = localStorage.getItem("valueFan");
document.getElementById('buttonSubmitFan').value = localStorage.getItem("valueFan");
function buttonEventFan() {
  if (localStorage.getItem("valueFan") === "Fan On") localStorage.setItem("valueFan", "Fan Off");
  else localStorage.setItem("valueFan", "Fan On");
}
```

Pentru a nu pierde informațiile ținute de site în momentul unui “refresh” sau în urma unui eveniment de tipul POST/GET, vom stoca valorile curente, local, în memoria browser-ului prin intermediul unor „variabile” (asemănător unui cookie). Valorile primite de la aplicația web vor fi prelucrate de către server și se vor efectua anumite acțiuni corespunzătoare valorilor primite.

Primirea informațiilor de la server pentru a le afișa pe site se realizează prin intermediul unei funcții ce trimite o cerere către server, iar răspunsul serverului va fi pachetul de date ce conține toate informațiile care vor fi afișate pe site, după o anumită structură (“Data1\nData2\n....DataN\n”).

```
document.getElementById("temperature-level").innerHTML = "Temperature" + "<br>" + localStorage.getItem("DataValue0") + "&degC";
document.getElementById("humidity-level").innerHTML = "Humidity" + "<br>" + localStorage.getItem("DataValue1") + "%";
document.getElementById("heat-index").innerHTML = "Real feel" + "<br>" + localStorage.getItem("DataValue2") + "&degC";
document.getElementById("photo-R-level").innerHTML = "Photo-R level" + "<br>" + localStorage.getItem("DataValue3");
document.getElementById("locked-unlocked-status").innerHTML = localStorage.getItem("DataValue4");
document.getElementById("driver-seat-status").innerHTML = "Driver seat" + "<br>" + localStorage.getItem("DataValue5");

function updateData() {
    let request = new XMLHttpRequest();
    request.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            DataValue = this.responseText.split('\n');
            localStorage.setItem("DataValue0", DataValue[0]);
            localStorage.setItem("DataValue1", DataValue[1]);
            localStorage.setItem("DataValue2", DataValue[2]);
            localStorage.setItem("DataValue3", DataValue[3]);
            localStorage.setItem("DataValue4", DataValue[4]);
            localStorage.setItem("DataValue5", DataValue[5]);
        }
    };
    request.open('GET', '/data.txt', true);
    request.send();
}
```

Ca și în cazul cererilor de tip POST, vom stoca în memoria browser-ului valorile primite sub forma unui vector și vor fi afișate corespunzător.