WEB-APP

```jsx
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import './App.css';
import Stats from '../Stats/Stats';
import About from '../About/About';

function App() {
  return (
    <div className='App'>
      <Router>
        <Routes>
          <Route exact path='/' element={<Stats />} />
          <Route exact path='/about' element={<About />} />
        </Routes>
      </Router>
    </div>
  );
}

export default App;
```

Aplicatia Web are la baza fisierului App care initializeaza 2 sectiuni cea de Stats unde sunt puse datele din statie si cea de About Us care descrie membri echipei.

```jsx
import React from 'react';
import { Link } from 'react-router-dom';
import './Header.css';

function Header() {
  return (
    <div className='Header'>
      <div className='Nav-Buttons'>
        <Link to='/' style={{ textDecoration: 'none' }}>
          <div className='Nav-Button'>Stats</div>
        </Link>
        <Link to='/about' style={{ textDecoration: 'none' }}>
          <div className='Nav-Button'>About Us</div>
        </Link>
      </div>
    </div>
  );
}

export default Header;
```

In fisierul Header am facut si stilizat partea de sus a pagini de unde putem selecta daca dorim sa vedem datele sau descrierile despre echipa.

```
11  function Stats() {
12    const [temps, setTemp] = useState([]);
13    const [times, setTime] = useState([]);
14    const [pressures, setPressure] = useState([]);
15    const [winds, setWind] = useState([]);
16    const [humiditys, setHumidity] = useState([]);
17
18    useEffect(() => {
19      fetchData();
20    }, []);
21
22    const fetchData = async () => {
23      try {
24        const api_url_latest =
25          'https://awu4j6hku3.execute-api.eu-central-1.amazonaws.com/dev/weather/latest';
26        const api_url =
27          'https://awu4j6hku3.execute-api.eu-central-1.amazonaws.com/dev/weather/hist?days=1&group=1';
28
29        const response = await fetch(api_url);
30
31        if (!response.ok) {
32          throw new Error('Failed to fetch weather data');
33        }
34
35        const data = await response.json();
36
37        const temp = [];
38        const humidity = [];
39        const wind = [];
40        const pressure = [];
```

Fisierul Stats contine o functie asincrona care ne ajuta sa facem rost de datele din api cu ajutorul functie fetch, aceste date se atribuie variabilei “data” fiind de tip obiect.

```
43          for (const item of data) {
44            if (item.dataType === 'temp') {
45              temp.push(item.value);
46            } else if (item.dataType === 'humidity') {
47              humidity.push(item.value);
48            } else if (item.dataType === 'wind') {
49              wind.push(item.value);
50            } else if (item.dataType === 'pressure') {
51              pressure.push(item.value);
52            }
53
54            const formattedTime = new Date(item.time).toLocaleTimeString('en-US', {
55              hour12: false,
56              hour: '2-digit',
57              minute: '2-digit',
58              second: '2-digit',
59            });
60
61            if (!time.includes(formattedTime)) {
62              time.push(formattedTime);
63            }
64
65            setTemp(temp);
66            setTime(time);
67            setPressure(pressure);
68            setHumidity(humidity);
69            setWind(wind);
70          }
```

Apoi din “data” introducem in vectori valorile.

```
        }
      } catch (error) {
        console.error('Error fetching data:', error);
      }
    };

    return (
      <div className='Stats'>
        <Header />
        <header className='Stats-Header'>
          <p className='Header-Text'>Weather Station Data</p>
        </header>
        <TemperatureChart data={temps} time={times} />
        <PressureChart data={pressures} time={times} />
        <WindChart data={winds} time={times} />
        <HumidityChart data={humiditys} time={times} />
        <Footer />
      </div>
    );
  }

  export default Stats;
```

La final transmitem vectorii cu datele, fisierelor care ne vor afisa graficele de tip chart.

```jsx
function TemperatureChart({ data, time }) {
  if (!Array.isArray(time)) {
    return null;
  }

  // Create an array of objects with 'time' and 'temperature' properties
  const chartData = time.map((t, index) => ({
    time: t,
    temperature: data[index],
  }));

  return (
    <div className='TemperatureChart'>
      <p className='title'>Temperature</p>
      <LineChart width={800} height={400} data={chartData}>
        <CartesianGrid strokeDasharray='3 3' />
        <XAxis dataKey='time' />
        <YAxis />
        <Tooltip />
        <Legend />
        <Line type='natural' dataKey='temperature' stroke='#c9a326' />
      </LineChart>
    </div>
  );
}

export default TemperatureChart;
```

Fisierul de Temperatura este aflat in exemplu restul sunt la fel, schimbate doar cateva nume de variabile si culorile acestora. Primim datele si verificam daca exista si nu sunt goale apoi facem niste operatii de map-are, i-ar mai apoi atribuim un titlu si obtiuni din libraria “recharts” cum ar fi legenda, marimea, data, culori si asa mai departe.

Fisierul de About Us nu contine nimic special doar text si stilizare.