



# Final project – IoT & Web

Weather Station UI

Kristian Sikanen

Harjoitustyö  
Huhtikuu 2023

Tietotekniikka  
TIETO22B

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sikanen, Kristian:  
Final project – IoT & Web  
Weather Station UI

Harjoitustyö 38 sivua, joista liitteitä 24 sivua  
Huhtikuu 2023

---

### Harjoitustyön ennakkotiedot

IoT & Web- kurssin päätteeksi toteutimme harjoitustyön, jossa toimeksiantona oli luoda HTML- pohjainen käyttöliittymä. Data haetaan TAMK:in omasta API:sta, esitellään HTML- sivustolla, ja käsitellään javascriptillä. HTML- sivuston muotoilussa kolmannen osapuolen kirjastot ja palvelut (mm. bootstrap) ovat sallittuja. Projektityö jaettu viiteen eri vaiheeseen, josta jokaisen vaiheen saavuttaminen takaa tietyn määrän pisteitä. Kokonaispistemäärä on 72p.

Ensimmäisen vaiheen tavoite on luoda esityskeino API:sta haettavalle datalle. Vaatimuksena luoda 3 eri HTML- tiedostoa, jossa jokaisessa esitetään tietty määrä API:in sijoitettuja mittapisteitä.

Vaiheessa kaksi työhön implementoidaan datan visualisointiin tarkoitettu työkalu chart.js.

Vaiheessa kolme dataa käsitellään javascriptissä, ja muodostetaan siitä 5 statistiikkaotetta. Funktiot ovat keskiarvo, mediaani, moodi, vaihteluväli ja keskihajonta.

Vaiheessa neljä työtä ja muodostettua koodia muokataan tavalla, joka mahdollistaa interaktiivisuuden käyttäjän ja ohjelman välillä. Aikaisemmat mittapisteiden visualisointikeinot muokataan nyt dynaamisiksi, ja määritellään datalle muuttuva tarkasteluväli.

---

## SISÄLLYS

1 Ohjelman rakenne & käytetyt työkalut.....	4
1.1 Ohjelman rakenne.....	4
1.2 Käytetyt työkalut.....	5
Itsearviointi.....	6
LIITTEET .....	7
Liite 1. Lähdekoodi – script.js.....	7
Liite 2. lähdekoodi – temperature.html .....	17
Liite 3. lähdekoodi – windspeed.html .....	21
Liite 4. lähdekoodi – index.html.....	25
Liite 5. lähdekoodi - last50readings.html.....	28
Liite 6. lähdekoodi - stylesheet.css .....	31
Liite 7. lähdekoodi - stylesheetIndex.css.....	33
Liite 8. lähdekoodi – scriptIndex.js .....	35
JOHDANTO	

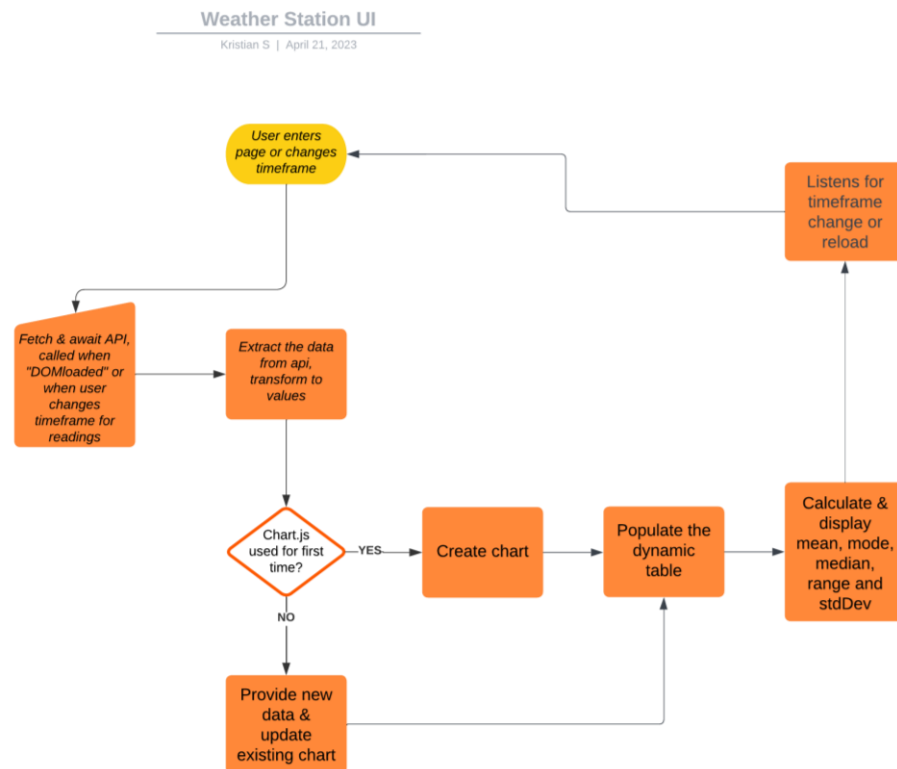
---

# 1 Ohjelman rakenne & käytetyt työkalut

## 1.1 Ohjelman rakenne

Projekti koostuu viidestä eri HTML- tiedostosta. Kaikki HTML- tiedostot jakavat yhden yleisen script.js- tiedoston, jossa kaikki toiminnallisuudet sivuille määritellään. Muotoilu ja sivujen ulkoasu on tuotettu pääosin käyttäen bootstrappia ja yhteistä stylesheet.css- tiedostoa. Viidestä sivusta kolme käyttää chart.js- palvelua.

kuva 1. HTML- tiedostot dynaamisesti rakentava javascript- funktio esitettynä flowchartissa.



HTML- sivuista kaikki toteuttavat kuvanmukaisen prosessin, jossa javascriptillä luodaan taulukko, sekä täytetään pöytä tiedolla. Käyttäjä saa itse myös valita aikaikkunan vaihtoehtoista live, 24-, 48- tai 72 tuntia. Valittavissa myös pidemmät ajanjaksot kuten 1 viikko, tai yksi kuukausi. Kun API on toimittanut pyydetyn tiedon, puretaan JSON- muodossa oleva data arvoiksi, ja sijoitetaan se muuttujaan jostamsen käyttö on mahdollista. Statistisissa laskutoimituksissa on hyödynnetty

mahdollisimman paljon aliohjelmia. Projekti on pyritty pitämään mahdollisimman modulaarisena ja yksiselitteisenä. (kts. lähdekoodin kommentointi)

## 1.2 Käytetyt työkalut

Projektissa käytetyt kolmansien osapuolien tuottamat palvelut & toiminnallisuudet:

- Chart.js
- Bootstrap
- gsap

Perusmuotoilu on suoritettu bootstrapin avulla, ja omalla stylesheetillä pyritty hienon lisäämään sivun ulkoista puoleensavetävyyttä. Gsap- animointikirjasto tarjoaa hienon mahdollisuuden lisätä sivuille näyttävyyttä otsikkotason animoinnin muodossa. Chart.js tarjoaa datan visualisointiin oivat työkalut.

Kyseessä on frontend- projekti, joten panostin erityisesti projektin ulkoasuun, käytettävyyteen ja datan esittämiseen puoleensavetävällä tavalla. Loin leikkisän väriteeman, suunnittelin layoutin etukäteen, ja toteutin visioni. Käytettävyyttä taas tukee sivujen täysi responsiivisuus, joka on saavutettu flexboxin, tai custom-breakpointtien avulla.

Integroin aikaisemman "forecast"- projektin tämän projektin etusivuksi. Nyt projekti on kokonaisuus joka näyttää tämänhetkisen sään, ja josta pääsee myös tutkimaan mittapisteitä ja statistiikkaa.

## Itsearviointi

Olen todella tyytyväinen työn lopputulokseen. Opin tämän lopputyön aikana todella laajasti javascriptiä. Käytin kokonaisuudessaan projektiin n. 30h koulun ulkopuolista aikaa. Olisin voinut ehkä jakaa ohjelmaa vielä hieman enemmän modulaarisemmaksi jakamalla yleisestä script.js tiedostosta jokaisen HTML- sivuston omat toiminnot niille suunnattuun omaan .js tiedostoon. Kompastelin myös työn aikana joidenkin funktioiden luomisessa. Kokonaisuutena olen kuitenkin myös koodipuoleeni todella tyytyväinen, ottaen huomioon että tämä on ensimmäinen kurssi jossa olen törmännyt javascriptiin, HTML:ään ja CSS:ään.

Toteutin viidestä mahdollisesta vaiheesta neljä. Loin erilliset HTML- tiedostot, joissa datan sijoittelupaikat alustetaan ja muotoillaan käyttäjäystävällisesti. Käännytään javascriptin puoleen, ja täytetään paikat dynaamisesti tiedolla. Toin chart.js- kaaviot, ja sijoitin datan niihin onnistuneesti. Muotoilin koodiani uudelleen, jotta käyttäjä pystyisi vaiheen neljä mukaisesti valitsemaan datalle aikaikkunan. Projekti on julkaistu ja työnnetty github pagesiin. Näinollen olisin oikeutettu 67-72 pisteeseen, joten arvioisin työni arvosanaksi 5.

## LIITTEET

### Liite 1. Lähdekoodi – script.js

Lähdekoodit myös saatavilla githubista, osoitteessa <https://github.com/Krisbis/WeatherApiUI>

```
console.log("script.js linked");
chartEmpty = true;
let chart = null;

// -----APIS-----
// const lastFiftyAPI =
//   "http://webapi19sa-1.course.tamk.cloud/v1/weather/limit/50";
// const tempTwentyAPI =
//   "http://webapi19sa-1.course.tamk.cloud/v1/weather/temperature";
// const winSpdTwentyAPI =
//   "http://webapi19sa-1.course.tamk.cloud/v1/weather/wind_speed";
// -----

async function fetchLastfifty(table) {
  try {
    const lastFiftyAPI =
      "http://webapi19sa-1.course.tamk.cloud/v1/weather/limit/50";
    const response = await fetch(lastFiftyAPI);
    const apiData = await response.json();

    // Extract the types, values and time from API
    // Types and values were in nested object, unlike the individual
    // value holders, like temperature or windspeed
    // so i iterated through all of the objects called "data"
    // and pushed them to an array created outside of the function scope
    const times = apiData.slice(-50).map((data) => data.date_time);
    const dataArr = [];
    apiData.forEach(function (obj) {
      for ([type, value] of Object.entries(obj.data)) {
        dataArr.push({ type, value });
      }
    });

    // Create the table of information
    const tableHead = table.querySelector("thead");
    const tableBody = table.querySelector("tbody");
    tableHead.innerHTML =
      '<tr><th scope="col">#</th><th scope="col">Time & Date</th><th scope="col">Type</th><th scope="col">Value</th></tr>';
  }
}
```

```

tableBody.innerHTML = "";
for (let i = 1; i < times.length; i++) {
  const t = times[i];
  const type = dataArr[i].type;
  const value = dataArr[i].value;
  const cellElement = document.createElement("tr");
  cellElement.innerHTML =
    '<td scope="row">' +
    i +
    "</td><td>" +
    convertTime(t) +
    "</td><td>" +
    type +
    "</td><td>" +
    value +
    "</td>";
  tableBody.appendChild(cellElement);
}
} catch (error) {
  console.log(error);
}
}

// Conditional call for the function - event listener calls
// function directly if the filepath returns true
if (window.location.href.endsWith("/last50readings.html")) {
  document.addEventListener("DOMContentLoaded", function () {
    fetchLastfifty(document.querySelector("table"));
  });
}

async function fetchTempTwenty(table, limit) {
  try {
    const tempTwentyAPI = `https://webapi19sa-
1.course.tamk.cloud/v1/weather/temperature/${limit}`;
    const response = await fetch(tempTwentyAPI);
    const apiData = await response.json();

    // Extract the temperature values from the API data
    const temperatures = apiData.map((data) => data.temperature);
    const times = apiData.map((data) => data.date_time);

    // Create the Chart.js chart
    if (chartEmpty) {
      type = "Temperature";
      makeChart(temperatures, limit, type);
      chartEmpty = false;
    } else {
      chart.data.datasets[0].data = temperatures;
      chart.data.labels = [...Array(times.length).keys()].map(
        (i) => `Reading ${i + 1}`
      );
    }
  }
}

```



```

    chart.update();
  }

  // Create the table of information
  const tableHead = table.querySelector("thead");
  const tableBody = table.querySelector("tbody");
  const tableDesc = document.getElementById("apiLog");
  tableDesc.innerHTML = `API readings from timeframe: ${limit}`;
  tableHead.innerHTML =
    '<tr><th scope="col">#</th><th scope="col">Time & Date</th><th scope="col">Temperature</th></tr>';
  tableBody.innerHTML = "";
  for (let i = 1; i < times.length; i++) {
    const t = times[i];
    const c = temperatures[i];
    const cellElement = document.createElement("tr");
    cellElement.innerHTML =
      '<td scope="row">' +
      i +
      '</td><td>' +
      convertTime(t) +
      '</td><td>' +
      c +
      " °C" +
      '</td>';
    tableBody.appendChild(cellElement);
  }

  // Statistic- variables to point html elements,
  // then assigning values to them by external function
  // Would have done other js- file specifically for statistics,
  // but these are easy to implement so i will keep them in main script
  const temperaturesArr = roundToOneDecimal(temperatures);
  const meanElement = document.getElementById("mean");
  const modeElement = document.getElementById("mode");
  const medianElement = document.getElementById("median");
  const rangeElement = document.getElementById("range");
  const stdDevElement = document.getElementById("stdDev");
  meanElement.innerHTML = getMean(temperaturesArr) + "°C";
  modeElement.innerHTML = getMode(temperaturesArr) + "°C";
  medianElement.innerHTML = getMedian(temperaturesArr) + "°C";
  rangeElement.innerHTML = getRange(temperaturesArr) + "°C";
  stdDevElement.innerHTML = getStdDev(temperaturesArr) + "°C";
} catch (error) {
  console.log(error);
}
}

// Conditional call for the function
if (window.location.href.endsWith("/temperature.html")) {
  const table = document.querySelector("table");
  const limitInput = document.getElementById("limitInput");

```

```

limitInput.addEventListener("change", () => {
  const limit = limitInput.value;
  console.log(limit);
  fetchTempTwenty(table, limit);
});
// Default call with limit of 20
document.addEventListener("DOMContentLoaded", function () {
  const defaultLimit = 20;
  fetchTempTwenty(table, defaultLimit);
});
}

async function fetchWinSpdTwenty(table, limit) {
  try {
    const winSpdTwentyAPI = `http://webapi19sa-
1.course.tamk.cloud/v1/weather/wind_speed/${limit}`;
    const response = await fetch(winSpdTwentyAPI);
    const apiData = await response.json();

    // Extract the temperature values from the API data
    const windspeeds = apiData.map((data) => data.wind_speed);
    const times = apiData.map((data) => data.date_time);

    // Create the Chart.js chart
    if (chartEmpty) {
      type = "Windspeed";
      makeChart(windspeeds, limit, type);
      chartEmpty = false;
    } else {
      chart.data.datasets[0].data = windspeeds;
      chart.data.labels = [...Array(times.length).keys()].map(
        (i) => `Reading ${i + 1}`
      );
      chart.update();
    }

    // Create the table of information
    const tableHead = table.querySelector("thead");
    const tableBody = table.querySelector("tbody");
    const tableDesc = document.getElementById("apiLog");
    tableDesc.innerHTML = `API readings from timeframe: ${limit}`;
    tableHead.innerHTML =
      '<tr><th scope="col">#</th><th scope="col">Time & Date</th><th
scope="col">Windspeed</th></tr>';
    tableBody.innerHTML = "";
    for (let i = 1; i < times.length; i++) {
      const t = times[i];
      const c = windspeeds[i];
      const cellElement = document.createElement("tr");
      cellElement.innerHTML =

```

```

        '<td scope="row">' +
        i +
        "</td><td>" +
        convertTime(t) +
        "</td><td>" +
        c +
        " m/s" +
        "</td>";
        tableBody.appendChild(cellElement);
    }

    // Statistic- variables to point html elements,
    // then assigning values to them by external function
    // Would have done other js- file specifically for statistics,
    // but these are easy to implement so i will keep them in main script
    const windspeedsArr = roundToOneDecimal(windspeeds);
    const meanElement = document.getElementById("mean");
    const modeElement = document.getElementById("mode");
    const medianElement = document.getElementById("median");
    const rangeElement = document.getElementById("range");
    const stdDevElement = document.getElementById("stdDev");
    meanElement.innerHTML = getMean(windspeedsArr) + " m/s";
    modeElement.innerHTML = getMode(windspeedsArr) + " m/s";
    medianElement.innerHTML = getMedian(windspeedsArr) + " m/s";
    rangeElement.innerHTML = getRange(windspeedsArr) + " m/s";
    stdDevElement.innerHTML = getStdDev(windspeedsArr) + " m/s";
  } catch (error) {
    console.log(error);
  }
}

if (window.location.href.endsWith("/windspeed.html")) {
  const table = document.querySelector("table");
  const limitInput = document.getElementById("limitInput");

  limitInput.addEventListener("change", () => {
    const limit = limitInput.value;
    console.log(limit);
    fetchWinSpdTwenty(table, limit);
  });
  // Default call with limit of 20
  document.addEventListener("DOMContentLoaded", function () {
    const defaultLimit = 20;
    fetchWinSpdTwenty(table, defaultLimit);
  });
}

async function fetchHumidity(table, limit) {
  try {
    const humidityAPI = `https://webapi19sa-
1.course.tamk.cloud/v1/weather/humidity_out/${limit}`;
    const response = await fetch(humidityAPI);

```

```

const apiData = await response.json();

// Extract the temperature values from the API data
const humidities = apiData.map((data) => data.humidity_out);
const times = apiData.map((data) => data.date_time);

// Create the Chart.js chart
if (chartEmpty) {
  type = "Humidity";
  makeChart(humidities, limit, type);
  chartEmpty = false;
} else {
  chart.data.datasets[0].data = humidities;
  chart.data.labels = [...Array(times.length).keys()].map(
    (i) => `Reading ${i + 1}`
  );
  chart.update();
}

// Create the table of information
const tableHead = table.querySelector("thead");
const tableBody = table.querySelector("tbody");
const tableDesc = document.getElementById("apiLog");
tableDesc.innerHTML = `API readings from timeframe: ${limit}`;
tableHead.innerHTML =
  '<tr><th scope="col">#</th><th scope="col">Time & Date</th><th scope="col">Humidity</th></tr>';
tableBody.innerHTML = "";
for (let i = 1; i < times.length; i++) {
  const t = times[i];
  const c = humidities[i];
  const cellElement = document.createElement("tr");
  cellElement.innerHTML =
    '<td scope="row">' +
    i +
    '</td><td>' +
    convertTime(t) +
    '</td><td>' +
    c +
    ' %' +
    '</td>';
  tableBody.appendChild(cellElement);
}

// Statistic- variables to point html elements,
// then assigning values to them by external function
// Would have done other js- file specifically for statistics,
// but these are easy to implement so i will keep them in main script
const humiditiesArr = roundToOneDecimal(humidities);
const meanElement = document.getElementById("mean");
const modeElement = document.getElementById("mode");

```

```

    const medianElement = document.getElementById("median");
    const rangeElement = document.getElementById("range");
    const stdDevElement = document.getElementById("stdDev");
    meanElement.innerHTML = getMean(humiditiesArr) + "%";
    modeElement.innerHTML = getMode(humiditiesArr) + "%";
    medianElement.innerHTML = getMedian(humiditiesArr) + "%";
    rangeElement.innerHTML = getRange(humiditiesArr) + "%";
    stdDevElement.innerHTML = getStdDev(humiditiesArr) + "%";
  } catch (error) {
    console.log(error);
  }
}

// Conditional call for the function
if (window.location.href.endsWith("/humidity.html")) {
  const table = document.querySelector("table");
  const limitInput = document.getElementById("limitInput");

  limitInput.addEventListener("change", () => {
    const limit = limitInput.value;
    console.log(limit);
    fetchHumidity(table, limit);
  });

  // Default call with limit of 20
  document.addEventListener("DOMContentLoaded", function () {
    const defaultLimit = 20;
    fetchHumidity(table, defaultLimit);
  });
}

// This was formulated by chatGPT,
// however i took the time to somewhat understand
// what this function stands for.
function convertTime(time) {
  const date = new Date(time);
  const day = date.getDate().toString().padStart(2, "0");
  const month = (date.getMonth() + 1).toString().padStart(2, "0");
  const year = date.getFullYear().toString();
  const hours = date.getHours().toString().padStart(2, "0");
  const minutes = date.getMinutes().toString().padStart(2, "0");
  const formattedTime =
`${hours}:${minutes} / ${day}.${month}.${year}`;
  return formattedTime;
}

// Chart.js function
function makeChart(temperatures, times, type) {
  const canvas = document.getElementById("myChart");
  chart = new Chart(canvas, {
    type: "line",
    data: {
      labels: [...Array(times).keys()].map((i) => `Reading ${i + 1}`),

```

```

        datasets: [
            {
                label: type,
                data: temperatures,
                backgroundColor: "cyan",
                borderColor: "cyan",
                borderWidth: 1,
            },
        ],
    },
    options: {
        scales: {
            yAxes: [
                {
                    ticks: {
                        beginAtZero: true,
                    },
                },
            ],
        },
    },
});
}

// Refresh- function for the website
function autoRefresh() {
    window.location = window.location.href;
}
setInterval("autoRefresh()", 500000);

// Function for gsap text animation - windspeed, temperature.html
function animatedText() {
    let tl = gsap.timeline({ repeat: -1 });
    tl.to("h1", 30, { backgroundColor: "-960px 0" });
}
document.addEventListener("DOMContentLoaded", function () {
    animatedText();
});

// ----- STATISTIC FUNCTIONS -----
// Median- function
function getMedian(arr) {
    const sortedArr = arr.sort((a, b) => a - b);
    const mid = Math.floor(sortedArr.length / 2);

    if (sortedArr.length % 2 === 0) {
        outcome = (sortedArr[mid - 1] + sortedArr[mid]) / 2;
        return outcome.toFixed(1);
    } else {
        return sortedArr[mid].toFixed(1);
    }
}

```

```

}

// Mode- function
function getMode(arr) {
  let mode = 0;
  let count = 0;

  for (let i = 0; i < arr.length; i++) {
    let currentCount = 0;
    for (let j = 0; j < arr.length; j++) {
      if (arr[j] === arr[i]) {
        currentCount++;
      }
    }
    if (currentCount > count) {
      mode = arr[i];
      count = currentCount;
    }
  }

  return mode.toFixed(1);
}

// Mean- function
function getMean(arr) {
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
  }
  let outcome = sum / arr.length;
  return outcome.toFixed(1);
}

// Range- function
function getRange(arr) {
  const max = Math.max(...arr);
  const min = Math.min(...arr);
  const range = max - min;
  return range.toFixed(1);
}

// Standard deviation- function
function getStdDev(arr) {
  const n = arr.length;
  const mean = arr.reduce((a, b) => a + b) / n;
  const stdDev = Math.sqrt(
    arr.map((x) => Math.pow(x - mean, 2)).reduce((a, b) => a + b) / n
  );
  return stdDev.toFixed(1);
}

```

```
// Function that converts string objects to numbers, and
// also rounds them for statistical calculations
function roundToOneDecimal(numbers) {
  const roundedNumbers = [];
  for (let i = 0; i < numbers.length; i++) {
    const number = parseFloat(numbers[i]);
    const rounded = number.toFixed(1);
    roundedNumbers.push(parseFloat(rounded));
  }
  return roundedNumbers;
}
```



## Liite 2. lähdekoodi – temperature.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta
      http-equiv="Content-Security-Policy"
      content="upgrade-insecure-requests"
    />

    <title>Document</title>

    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="./css/stylesheet.css" type="text/css" />
  </head>
  <body>
    <div class="hero">
      <nav
        class="navbar navbar-expand-sm navbar bg-transparent"
        id="custom_navbar_styles"
      >
        <a href="./index.html" class="navbar-brand mx-4 p-0">
          <div class="text-black-50 h2">Weather</div></a>
        >
        <button
          class="navbar-toggler navbar-light"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#main-navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div id="main-navigation" class="collapse navbar-collapse">
          <ul class="navbar-nav ms-auto text-center mx-2">
            <li class="nav-item">
              <a class="nav-link" href="./last50readings.html">Recent</a>
            </li>

```

```

        <li class="nav-item">
            <a class="nav-link" href="./windspeed.html">Windspeed</a>
        </li>

        <li class="nav-item">
            <a class="nav-link" href="./temperature.html">Tempera-
ture</a>
        </li>

        <li class="nav-item">
            <a class="nav-link" href="./humidity.html">Humidity</a>
        </li>
    </ul>
</div>
</nav>
</div>
<br />
<br />
<div class="d-flex justify-content-center">
    <h1 id="animated_text">Temperature</h1>
</div>
<br />
<br />
<br />
<br />
<div class="chart_container d-flex">
    <canvas id="myChart"></canvas>
</div>
<br>
<br>
<div class="d-flex justify-content-center">
    <label for="limitInput"></label>
    <select
        id="limitInput"
        name="limitInput"
        class="form-select form-select-sm w-25"
    >
        <option value="">live</option>
        <option value="24">24 hours</option>
        <option value="48">48 hours</option>
        <option value="78">78 hours</option>
        <option value="1week">1 week</option>
        <option value="1month">1 month</option>
    </select>
</div>
<br />
<br />
<h4 class="opacity-50 text-center">Key Statistics</h4>
<div class="container d-flex justify-content-evenly text-center">
    <div class="stat_container d-flex flex-column justify-content-
evenly">

```

```

        <h5>Mean</h5>
        <h4 id="mean">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>Mode</h5>
        <h4 id="mode">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>Median</h5>
        <h4 id="median">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>Range</h5>
        <h4 id="range">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>StdDev</h5>
        <h4 id="stdDev">404</h4>
    </div>
</div>
<br />
<br />
<h3 class="d-flex justify-content-center opacity-50" id="apiLog">
    Log
</h3>
<div class="container d-flex">
    <table class="table table-hover table-striped table-borderless">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Time & Date</th>
                <th scope="col">Temperature</th>
            </tr>
        </thead>
        <tbody>
            <!-- Dynamic table from JS -->
        </tbody>
    </table>
</div>

<script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-a1-
pha3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ENjd04Dr2bkBIFxQpe-
oTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"
    crossorigin="anonymous"
></script>

```

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.11.4/gsap.min.js"></script>  
<script src="./js/script.js"></script>  
</body>  
</html>
```

## Liite 3. lähdekoodi – windspeed.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta
      http-equiv="Content-Security-Policy"
      content="upgrade-insecure-requests"
    />

    <title>Document</title>

    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="./css/stylesheet.css" type="text/css" />
  </head>
  <body>
    <div class="hero">
      <nav
        class="navbar navbar-expand-sm navbar bg-transparent"
        id="custom_navbar_styles"
      >
        <a href="./index.html" class="navbar-brand mx-4 p-0">
          <div class="text-black-50 h2">Weather</div></a>
        >
        <button
          class="navbar-toggler navbar-light"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#main-navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div id="main-navigation" class="collapse navbar-collapse">
          <ul class="navbar-nav ms-auto text-center mx-2">
            <li class="nav-item">
              <a class="nav-link" href="./last50readings.html">Recent</a>
            </li>

```

```

        <li class="nav-item">
            <a class="nav-link" href="./windspeed.html">Windspeed</a>
        </li>

        <li class="nav-item">
            <a class="nav-link" href="./temperature.html">Tempera-
ture</a>
        </li>

        <li class="nav-item">
            <a class="nav-link" href="./humidity.html">Humidity</a>
        </li>
    </ul>
</div>
</nav>
</div>
<br />
<br />
<div class="d-flex justify-content-center">
    <h1 id="animated_text">Windspeed</h1>
</div>
<br />
<br />
<br />
<br />
<div class="chart_container d-flex">
    <canvas id="myChart"></canvas>
</div>
<br />
<br />
<div class="d-flex justify-content-center">
    <label for="limitInput"></label>
    <select
        id="limitInput"
        name="limitInput"
        class="form-select form-select-sm w-25"
    >
        <option value="">live</option>
        <option value="24">24 hours</option>
        <option value="48">48 hours</option>
        <option value="78">78 hours</option>
        <option value="1week">1 week</option>
        <option value="1month">1 month</option>
    </select>
</div>
<br />
<br />
<h4 class="opacity-50 text-center">Key Statistics</h4>
<div class="container d-flex justify-content-evenly text-center">
    <div class="stat_container d-flex flex-column justify-content-
evenly">

```

```

        <h5>Mean</h5>
        <h4 id="mean">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>Mode</h5>
        <h4 id="mode">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>Median</h5>
        <h4 id="median">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>Range</h5>
        <h4 id="range">404</h4>
    </div>
    <div class="stat_container d-flex flex-column justify-content-
evenly">
        <h5>StdDev</h5>
        <h4 id="stdDev">404</h4>
    </div>
</div>
<br />
<br />
<h3 class="d-flex justify-content-center opacity-50" id="apiLog">
    Last 20 readings from API
</h3>
<div class="container d-flex">
    <table class="table table-hover table-striped table-borderless">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Time & Date</th>
                <th scope="col">Temperature</th>
            </tr>
        </thead>
        <tbody></tbody>
    </table>
</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ENjd04Dr2bkBIFxQpe-
oTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"
    crossorigin="anonymous"
></script>

```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.11.4/gsap.min.js"></script>  
  <script src="./js/script.js"></script>  
</body>  
</html>
```



## Liite 4. lähdekoodi – index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

    <link
      href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=swap"
      rel="stylesheet"
    />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z" crossorigin="anonymous">
    <link rel="stylesheet" href="./css/stylesheetIndex.css" type="text/css" />
  </head>
  <body>
    <div class="hero">
      <nav
        class="navbar navbar-expand-sm navbar bg-transparent"
        id="custom_navbar_styles"
      >
        <a href="./index.html" class="navbar-brand mx-4 p-0">
          <div class="text-black-50 h2">Weather</div></a>
        >
        <button
          class="navbar-toggler navbar-light"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#main-navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div id="main-navigation" class="collapse navbar-collapse">
          <ul class="navbar-nav ms-auto text-center mx-2">
            <li class="nav-item">
              <a class="nav-link" href="./last50readings.html">Recent</a>
            </li>
            <li class="nav-item">

```

```

        <a class="nav-link" href="./windspeed.html">Windspeed</a>
    </li>

    <li class="nav-item">
        <a class="nav-link" href="./temperature.html">Tempera-
ture</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="./humidity.html">Humidity</a>
    </li>
</ul>
</div>
</nav>
</div>
<br>
<br>

<!-- I repurposed and modified earlier weather page project to be my
index page for this project, adds a bit flash and is useful-->
<div class="main_container">
    <h2 id="location_name">Tampere</h2>
    <h1 id="current_temp">°C</h1>

    <div id="weather_info">
        
        <h3 id="current_wind_speed">wind speed</h3>
        
        <h3 id="current_cloud_coverage">cloud coverage</h3>
        
        <h3 id="relative_humidity">humidity</h3>
    </div>
    <div id="weather_desc">kuvaus säästä</div>
    <div id="upcoming_weather">
        <span class="card" id="card1">
            <div id="time1">Time</div>
            <div id="temp1">temp</div>
            <div id="desc1">desc</div>
        </span>
        <span class="card" id="card2">
            <div id="time2">Time</div>
            <div id="temp2">temp</div>
            <div id="desc2">desc</div>
        </span>
        <span class="card" id="card3">
            <div id="time3">Time</div>
            <div id="temp3">temp</div>
            <div id="desc3">desc</div>
        </span>
        <span class="card" id="card4">
            <div id="time4">Time</div>
            <div id="temp4">temp</div>

```

```
        <div id="desc4">desc</div>
      </span>
      <span class="card" id="card5">
        <div id="time5">Time</div>
        <div id="temp5">temp</div>
        <div id="desc5">desc</div>
      </span>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-EN-
jd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe" cros-
sorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.11.4/gsap.min.js"></script>
  <script src="./js/scriptIndex.js"></script>
</body>
</html>
```

## Liite 5. lähdekoodi - last50readings.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta
      http-equiv="Content-Security-Policy"
      content="upgrade-insecure-requests"
    />

    <title>Document</title>

    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="./css/stylesheet.css" type="text/css" />
  </head>
  <body>
    <div class="hero">
      <nav
        class="navbar navbar-expand-sm navbar bg-transparent"
        id="custom_navbar_styles"
      >
        <a href="./index.html" class="navbar-brand mx-4 p-0">
          <div class="text-black-50 h2">Weather</div></a>
        >
        <button
          class="navbar-toggler navbar-light"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#main-navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div id="main-navigation" class="collapse navbar-collapse">
          <ul class="navbar-nav ms-auto text-center mx-2">
            <li class="nav-item">
              <a class="nav-link" href="./last50readings.html">Recent</a>
            </li>

```

```

        <li class="nav-item">
            <a class="nav-link" href="./windspeed.html">Windspeed</a>
        </li>

        <li class="nav-item">
            <a class="nav-link" href="./temperature.html">Tempera-
ture</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="./humidity.html">Humidity</a>
        </li>
    </ul>
</div>
</nav>
<br />
<br />
<div class="d-flex justify-content-center">
    <h1 id="animated_text">Recent</h1>
</div>
<br />
<br />
<br />
<br />
<h3 class="d-flex justify-content-center opacity-50">
    Last 50 readings from API
</h3>
<div class="container d-flex">
    <table class="table table-hover table-striped table-borderless">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Time & Date</th>
                <th scope="col">Type</th>
                <th scope="col">Value</th>
            </tr>
        </thead>
        <tbody>
            <!-- Dynamic table from js -->
        </tbody>
    </table>
</div>
<script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ENjdO4Dr2bkBIFxQpe-
oTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"
    crossorigin="anonymous"
></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"
    flare.com/ajax/libs/gsap/3.11.4/gsap.min.js"></script>

```

```
    <script src="./js/script.js"></script>  
  </div>  
</body>  
</html>
```

## Liite 6. lähdekoodi - stylesheet.css

```
html,
body {
  height: 100%;
  margin: 0;
  background-color: white;
  color: black;
}

canvas {
  padding-left: 0;
  padding-right: 0;
  margin-left: auto;
  margin-right: auto;
  width: 50%;
}

table {
  padding-left: 0;
  padding-right: 0;
  margin-left: auto;
  margin-right: auto;
  width: 75%;
}

thead{
  background-color: #D1D119;
  color: white;
}

::-webkit-scrollbar{
  width: 7px;
}

::-webkit-scrollbar-track{
  background: white;
}

::-webkit-scrollbar-thumb{
  background: #D1D119;
}

::-webkit-scrollbar-thumb:hover{
  background: #a9a91e;
}
```

```

.table-hover thead tr:hover th, .table-hover tbody tr:hover td{
  background-color: #dfdf15;
  color:green;
}

.nav-item>a:hover {
  color: green;
}

.stat_container{
  width: 125px;
  height: 100px;
  box-shadow: 0 0px 12px 0 rgba(1, 1, 1, 0.751);
  margin: 10px;
  border-radius: 15%;
  color: grey;
}

.stat_container:hover{
  color: white;
  background-color: #D1D119;
}

#animated_text{
  background-image: url(https://cdn.pixa-
bay.com/photo/2017/03/05/18/08/texture-2119294__340.jpg);
  background-attachment: fixed;
  -webkit-text-fill-color: transparent;
  -webkit-background-clip: text;
  margin: 0px;
  padding: 0px;
  font-size: 100px;
}

@media screen and (max-width: 600px) {
  #animated_text{
    font-size: 70px;
  }
  canvas{
    width: 100%;
  }
}

@media screen and (max-width: 992px) {
  canvas{
    width: 80%;
  }
}

```



## Liite 7. lähdekoodi - stylesheetIndex.css

```
html {
  display: flex;
  height: 100%;
  min-width: 100%;
}

body {
  height: 100%;
  min-width: 100%;
  margin: 0px;
  padding: 0%;
  height: 100%;
  background-color: white;
}

h1 {
  font-size: 150px;
}

h2 {
  font-size: 50px;
  font-weight: 100;
  color: black;
  opacity: 50%;
}

.main_container {
  text-align: center;
  color: rgb(0, 0, 0);
}

#current_temp {
  background-image: url(https://cdn.pixa-
bay.com/photo/2017/07/03/20/17/abstract-2468874_960_720.jpg);
  background-attachment: fixed;
  -webkit-text-fill-color: transparent;
  -webkit-background-clip: text;
  margin: 0px;
  padding: 0px;
}

.secondary_container {
  display: flex;
  color: white;
```

```
justify-content: center;
align-items: center;
overflow-y: hidden;
}

#weather_info {
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 25px;
  margin: 25px;
  color: black;
  opacity: 50%;
}

.logos {
  height: 70px;
}

#weather_desc {
  font-size: 40px;
  color: black;
  opacity: 50%;
  margin-bottom: 50px;
  margin-top: 25px;
  padding: 0px;
}

#upcoming_weather {
  font-size: medium;
  color: black;
  opacity: 75%;
}

.card {
  display: inline-flex;
  flex-direction: column;
  justify-content: space-around;
  width: 125px;
  height: 100px;
  box-shadow: 0 0px 12px 0 rgba(1, 1, 1, 0.751);
  transition: 0.3s;
  margin: 10px;
  border-radius: 15%;
}

.card:hover {
  color: white;
  background-color: #40d915;
}
```

## Liite 8. lühdekoodi – scriptIndex.js

```

const API =
  "https://api.open-meteo.com/v1/forecast?latitude=61.50&longitude=23.79&hourly=temperature_2m,relativehumidity_2m,weathercode,cloud-cover,windspeed_10m&current_weather=true&timezone=auto";

let tl = gsap.timeline({ repeat: -1 });
tl.to("h1", 30, { backgroundColor: "-960px 0" });

var today = new Date();
var time = today.getHours();

async function fetchData() {
  try {
    const response = await fetch(API);
    const data = await response.json();
    console.table([
      data.current_weather.temperature,
      data.current_weather.windspeed,
      data.current_weather.weathercode,
      data.current_weather.time,
      data.hourly.cloudcover,
      data.hourly.relativehumidity_2m,
      data.hourly.weathercode,
    ]);
    document.getElementById("current_temp").innerText =
      `${data.current_weather.temperature}` + "°C";
    const weatherCode = data.current_weather.weathercode;
    const weatherDescription = weatherDescriptions[weatherCode];
    document.getElementById("weather_desc").innerText = `${weatherDescription}`;
    document.getElementById("current_wind_speed").innerText =
      `${data.current_weather.windspeed}` + " km/h";
    document.getElementById("current_cloud_coverage").innerText =
      `${data.hourly.cloudcover[time]}` + " %";
    document.getElementById("relative_humidity").innerText =
      `${data.hourly.relativehumidity_2m[time]}` + " %";
    document.getElementById("time1").innerText = `${time+1}` + ".00";
    document.getElementById("time2").innerText = `${time+2}` + ".00";
    document.getElementById("time3").innerText = `${time+3}` + ".00";
    document.getElementById("time4").innerText = `${time+4}` + ".00";
    document.getElementById("time5").innerText = `${time+5}` + ".00";
    document.getElementById("temp1").innerText = `${data.hourly.temperature_2m[time+1]}` + " °C";
  } catch (error) {
    console.error("Error fetching data:", error);
  }
}

```

```

        document.getElementById("temp2").innerText = `${data.hourly.temperature_2m[time+2]}` + " °C";
        document.getElementById("temp3").innerText = `${data.hourly.temperature_2m[time+3]}` + " °C";
        document.getElementById("temp4").innerText = `${data.hourly.temperature_2m[time+4]}` + " °C";
        document.getElementById("temp5").innerText = `${data.hourly.temperature_2m[time+5]}` + " °C";

        const weatherForecast1 = data.hourly.weathercode[time+1]
        const weatherForecast2 = data.hourly.weathercode[time+2]
        const weatherForecast3 = data.hourly.weathercode[time+3]
        const weatherForecast4 = data.hourly.weathercode[time+4]
        const weatherForecast5 = data.hourly.weathercode[time+5]

        document.getElementById("desc1").innerText = `${weatherDescriptions[weatherForecast1]}`;
        document.getElementById("desc2").innerText = `${weatherDescriptions[weatherForecast2]}`;
        document.getElementById("desc3").innerText = `${weatherDescriptions[weatherForecast3]}`;
        document.getElementById("desc4").innerText = `${weatherDescriptions[weatherForecast4]}`;
        document.getElementById("desc5").innerText = `${weatherDescriptions[weatherForecast5]}`;
    } catch (error) {
        console.error(error);
    }
}
}
fetchData();

const weatherDescriptions = {
    0: "Clear sky",
    1: "Clear sky",
    2: "Few clouds",
    3: "Scattered clouds",
    4: "Broken clouds",
    5: "Rain showers",
    6: "Rain showers",
    7: "Rain",
    8: "Rain and snow showers",
    9: "Snow showers",
    10: "Rain and hail showers",
    11: "Hail showers",
    12: "Light rain",
    13: "Rain and snow",
    14: "Light snow",
    15: "Heavy snow",
    16: "Thunderstorm",
    17: "Thunderstorm and rain",
    18: "Thunderstorm and snow",

```

```
19: "Thunderstorm and hail",
20: "Mist",
21: "Fog",
22: "Snow",
23: "Rain and sleet",
24: "Sleet",
25: "Freezing rain",
26: "Freezing drizzle",
27: "Freezing fog",
28: "Patches of fog",
29: "Shallow fog",
30: "Partial fog",
31: "Overcast",
32: "Cloudy",
33: "Mostly cloudy",
34: "Scattered clouds",
35: "Few clouds",
36: "Partly cloudy",
37: "Mostly clear",
38: "Clear",
39: "Blowing snow",
40: "Rain and thunderstorm",
41: "Snow and thunderstorm",
42: "Hail and thunderstorm",
43: "Rain, snow, and thunderstorm",
44: "Snow and fog",
45: "Hail",
46: "Snow and sleet",
47: "Rain, snow, and sleet",
48: "Light drizzle",
49: "Drizzle",
50: "Heavy drizzle",
51: "Light rain and snow",
52: "Rain and snow",
53: "Light snow showers",
54: "Snow showers",
55: "Heavy snow showers",
56: "Light snow blowing",
57: "Snow blowing",
58: "Heavy snow blowing",
59: "Hail showers",
60: "Light rain showers",
61: "Moderate rain showers",
62: "Heavy rain showers",
63: "Light rain and hail",
64: "Moderate rain and hail",
65: "Heavy rain and hail",
66: "Light rain and snow",
67: "Moderate rain and snow",
68: "Heavy rain and snow",
69: "Light snow and rain",
```

```
70: "Moderate snow and rain",
71: "Heavy snow and rain",
72: "Light sleet showers",
73: "Moderate sleet showers",
74: "Heavy sleet showers",
75: "Light snow and hail",
76: "Moderate snow and hail",
77: "Heavy snow and hail",
78: "Light sleet and snow",
79: "Moderate sleet and snow",
80: "Heavy sleet and snow",
81: "Light snow and sleet",
82: "Moderate snow and sleet",
83: "Heavy snow and sleet",
84: "Light rain and drizzle",
85: "Moderate rain and drizzle",
86: "Heavy rain and drizzle",
87: "Light rain and freezing rain",
88: "Moderate rain and freezing rain",
89: "Heavy rain and freezing rain",
90: "Light freezing rain",
91: "Moderate freezing rain",
92: "Heavy freezing rain",
93: "Light drizzle and freezing rain",
94: "Moderate drizzle and freezing rain",
95: "Heavy drizzle and freezing rain",
96: "Light snow and freezing rain",
97: "Moderate snow and freezing rain",
98: "Heavy snow and freezing rain",
99: "Light freezing rain and snow",
100: "Moderate or heavy freezing rain and snow",
};

function autoRefresh() {
    window.location = window.location.href;
}
setInterval('autoRefresh()', 500000);
```