Assignment 5 (6 points total)

In this assignment we will take a look at SVG, which will allow us to add dynamic images to our website, get familiar with caching to avoid downloading all the required files upon each request to our server and send some data from a microcontroller to our server over the internet using MQTT.

Prerequisites:

- Have Node and Express installed
- Install `multer` middleware for Express using NPM
- Install `mqtt` package using NPM

Task 1

In this task you will have to add a dynamic image of a thermometer to your website which will reflect the temperature you type on a page. The height of the gauge and the color of both the gauge and the bulb should change depending on the temperature, with the color being smoothly interpolated between blue for cold and red for warm. Both properties should be animated. If the temperature on the page is not changed within the last 10 seconds, the thermometer should display a random temperature each second, otherwise it should display the user set temperature. Choose a reasonable temperature range, i.e. from -30° to +30°.

Hints:

- Inspect the provided SVG. The objects you should modify have an ID set on them
- Use `<object>` to add an SVG to your page
- Use `clearInterval()` to cancel the scheduled callback

Task 2

For the second task you will have to create a simple image gallery where you can upload your own files. Your server must also instruct your browser to cache the images and **not** revalidate image files upon request to improve loading speeds. Right-clicking on the image should delete it from the gallery.

Hints:

- Use `enctype="multipart/form-data"` on your form to send files
- Use `Cache-Control` HTTP header to tell the browser to cache images

Task 3

The purpose of this task is to use MQTT to be able to get temperature data. This week’s exercise will take advantage of a python script to publish the temperature data to your server.

In order to test the functionality of your application, you will need to install the following applications:

- Mosquitto (MQTT Broker Software)
- Mosquitto client (MQTT Client Software)
- Python3.9 (In order to run the script)

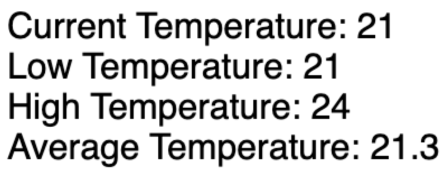
In the mosquitto.conf file (created on your mosquitto install), you will need to add two lines to access it from outside of your localhost.

- listener <port number>
- allow_anonymous true

The gist of this assignment is to use MQTT to subscribe to temperature sensor data that has been published by the included python script. The data will come in JSON format and will follow the following format.

```
{
	"samplenr": 3536,
	"timestamp": 3539372,
	"temperature": -53
}
```

Your output should appear like the following as well as storing your data in a database.



```
Current Temperature: 21
Low Temperature: 21
High Temperature: 24
Average Temperature: 21.3
```

Your server will need to use MQTT to subscribe to messages on the topic, “sim/sensors/tmp”

You will also need to modify the ip and port variables in the script to work with your MQTT broker.

Hint:

- npm i mqtt

IMPORTANT INFORMATION

Show all of your sources to the teacher before the deadline for full points. Assignments submitted even a moment after the deadline will only be able to get half points.

!!!There is no exception to this rule!!!