Assignment 6 (6 points total)

In this assignment we will work with property descriptors in JavaScript, see JS UI frameworks in action and take a look at WebSocket to make our pages even more interactive.

Prerequisites:

- Have Node and Express installed
- Install `ws` package using NPM

Task 1

For this task you will have to write a library to simplify saving and loading data from a file. Your library should make use of a constructor function (it should have **no return value**) which returns an object with at least two methods: `load()`, which loads the data from file and `save()`, which saves the data to the file. All other properties of the object should be modifiable, like in a regular object. These two special functions should be protected from deletion, modification and not be listed in loops. Add automatic JSON serialization upon saving and de-serialization during loading. You may also add a way to choose the file path to use.

Hints:

- Use `Object.defineProperty()` and `Object.defineProperties()` to set properties and their descriptors
- Use `Object.getOwnPropertyDescriptor()` to check whether descriptors have been set correctly
- Call a constructor function with the “`new`” keyword

Task 2

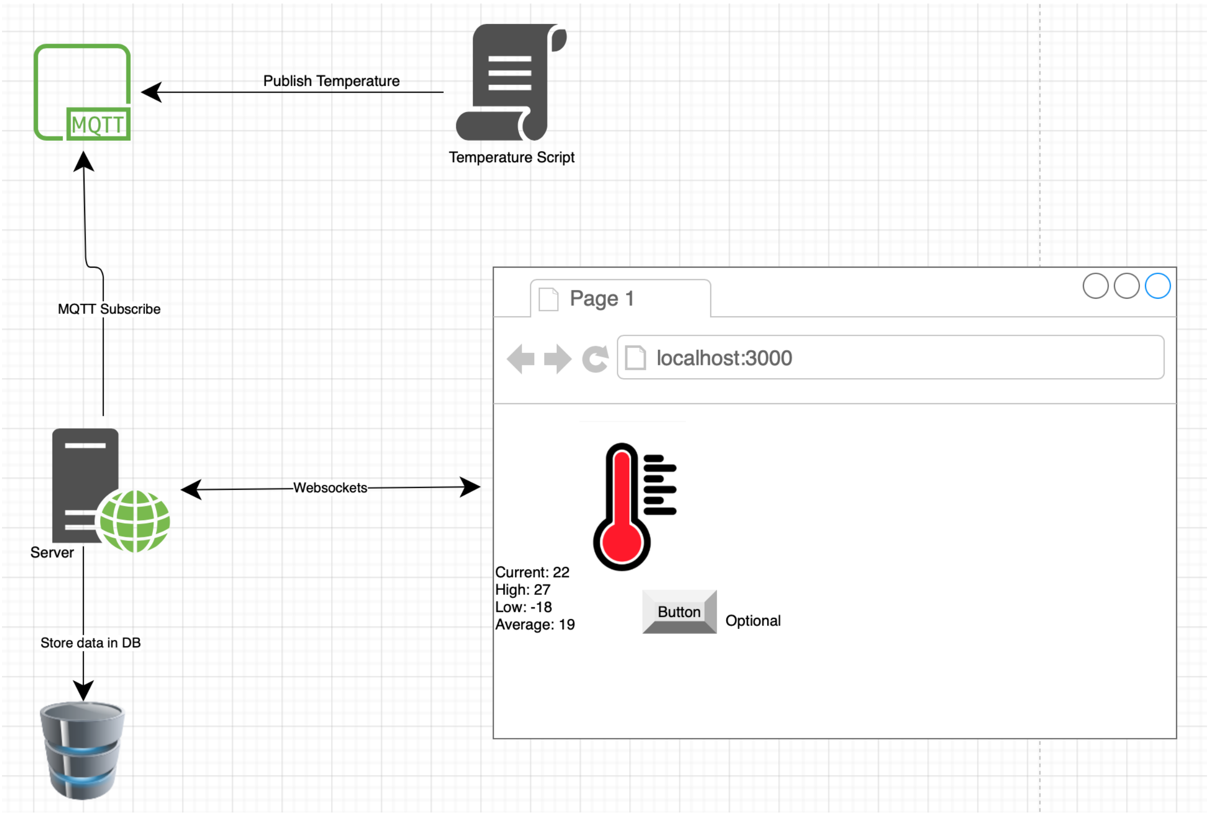
In this task you will have to create an interactive web application for tracking To-Do items. You must be able to view items already added as well as add new items to the list. The items themselves must have an option to be marked as done. The state of the list should be saved on the browser so that the list persists upon page refresh. Use a framework of your choice to implement this task.

Hints:

- Use `localStorage` to save your list on the browser
- You don’t necessarily need Node/Express for this task

Task 3

For this task we will get familiar with websockets. In this assignment we will look at getting temperature through MQTT again. For this you may use the js script from exercise 5. The idea here is to collect the temperature data and store it into a database as we have previously. The difference in this case is that you will need to send the current temperature through Websockets to display it in real-time on the webpage. Additionally you should display the temperature in real-time on the thermometer SVG that was provided last week.



The aggregate data can be displayed upon loading the page or whenever it is reloaded. Optionally you could request data through websockets and serve the aggregate data in real-time.

For the Websockets interface the ws package was used. This can be installed via npm.

IMPORTANT INFORMATION

Show your assignment 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!!!