**Website manual**

Contents

[Functional diagram 2](#_Toc487623659)

[Login page 3](#_Toc487623660)

[Sensor/Actuator page 4](#_Toc487623661)

[Dashboard 5](#_Toc487623662)

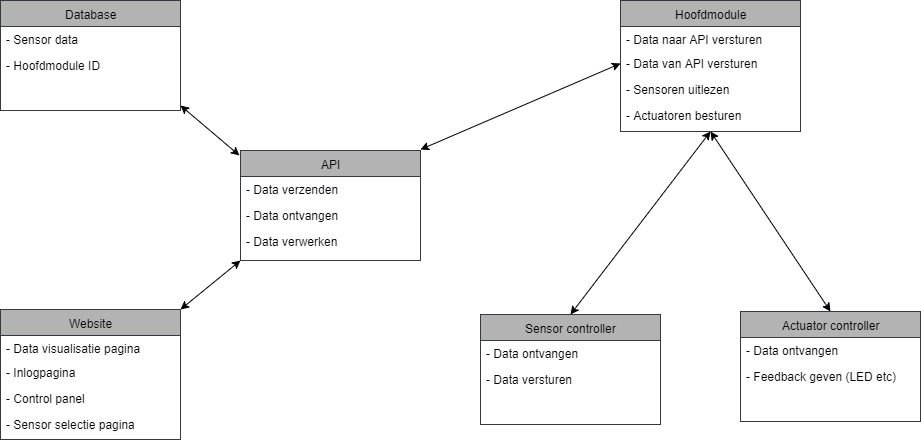
[Configuration page 5](#_Toc487623663)

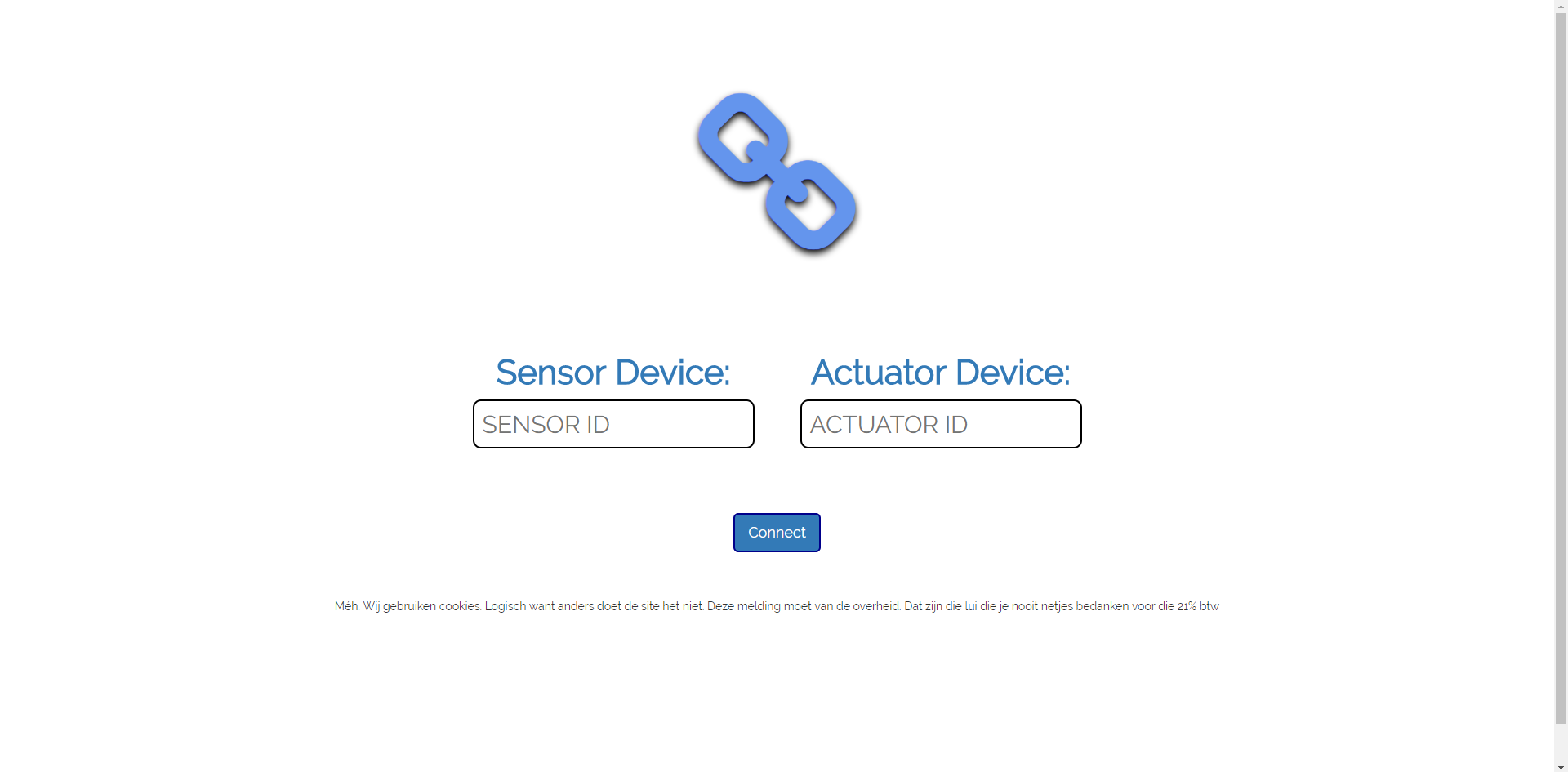
[Visualization page 6](#_Toc487623664)

[Bootstrap framework 7](#_Toc487623665)

[Code documentation 7](#_Toc487623666)

# Functional diagram

The website is supposed to communicate through the API with the Database. This way the website can fetch data from the database to show it to the user. The website communicates with the API with PHP scripts. There are different data types the website sends and receives. The website receives the sensor data and visualizes this on the data visualization page. The websites sends the threshold for the actuator to the API which in turn sends it to the database and the nodeMCU module. The API also manages the cookies saved on the client side. This way the website can “remember” some of the clients settings like the sensor and actuator modules selected, the nodeMCU IDs and the threshold. We used cookies because this was the easiest way to save and delete data of different users quickly.

Login page

<https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/LoginPage.php>

The login page serves for connection two devices with eachother. This is done by sending a POST request of the input form to the API. This purpose of this is to link the two devices with eachother in the database.

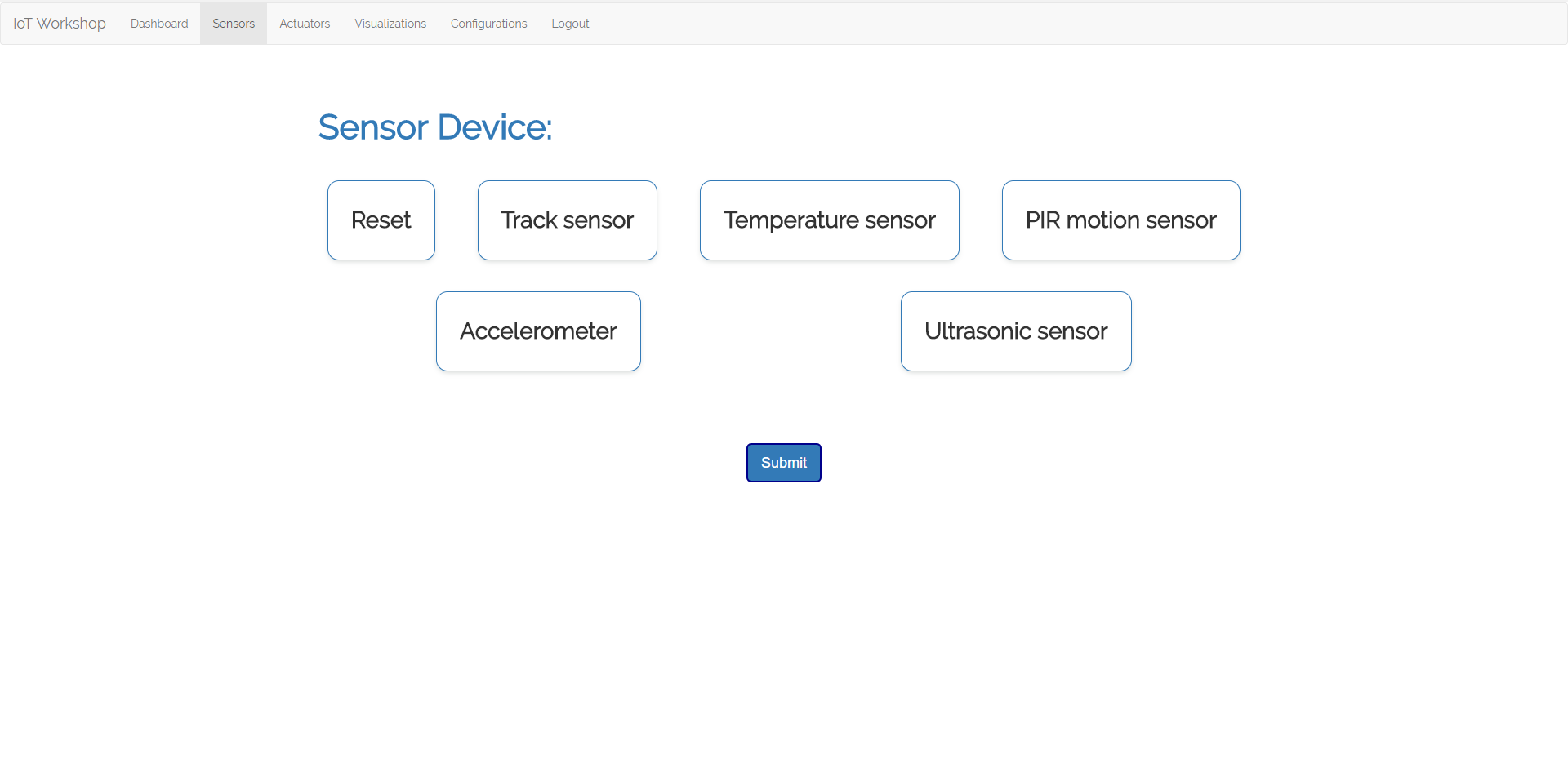
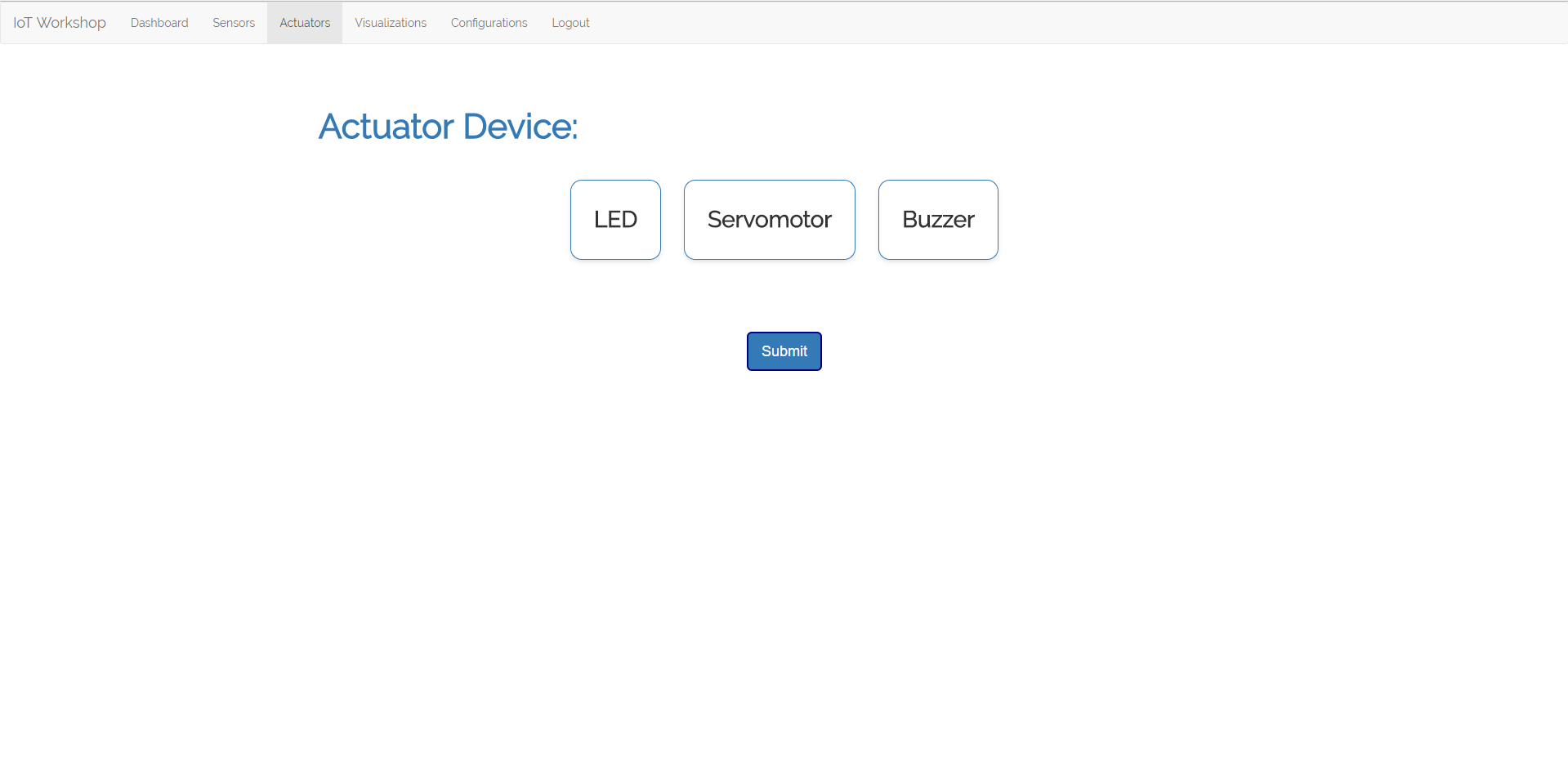
The page contains:

A Logo: The “link” symbol was designed to make it clear that the user is currently linking his devices and to improve the user experience.

An input form for the Device IDs: The input form sends a POST request to the API as soon as the connect button is pressed.

A connect button: The connect buttons sends a POST to the API with the Device IDs. The API handles the POST and links the devices in the database.

Error messages: An error message, depending on the error, is generated with a PHP switch statement which gets the error message of the API. The error message gets echoed to the login page and is displayed to the user. This way the user can contact us to solve their error.

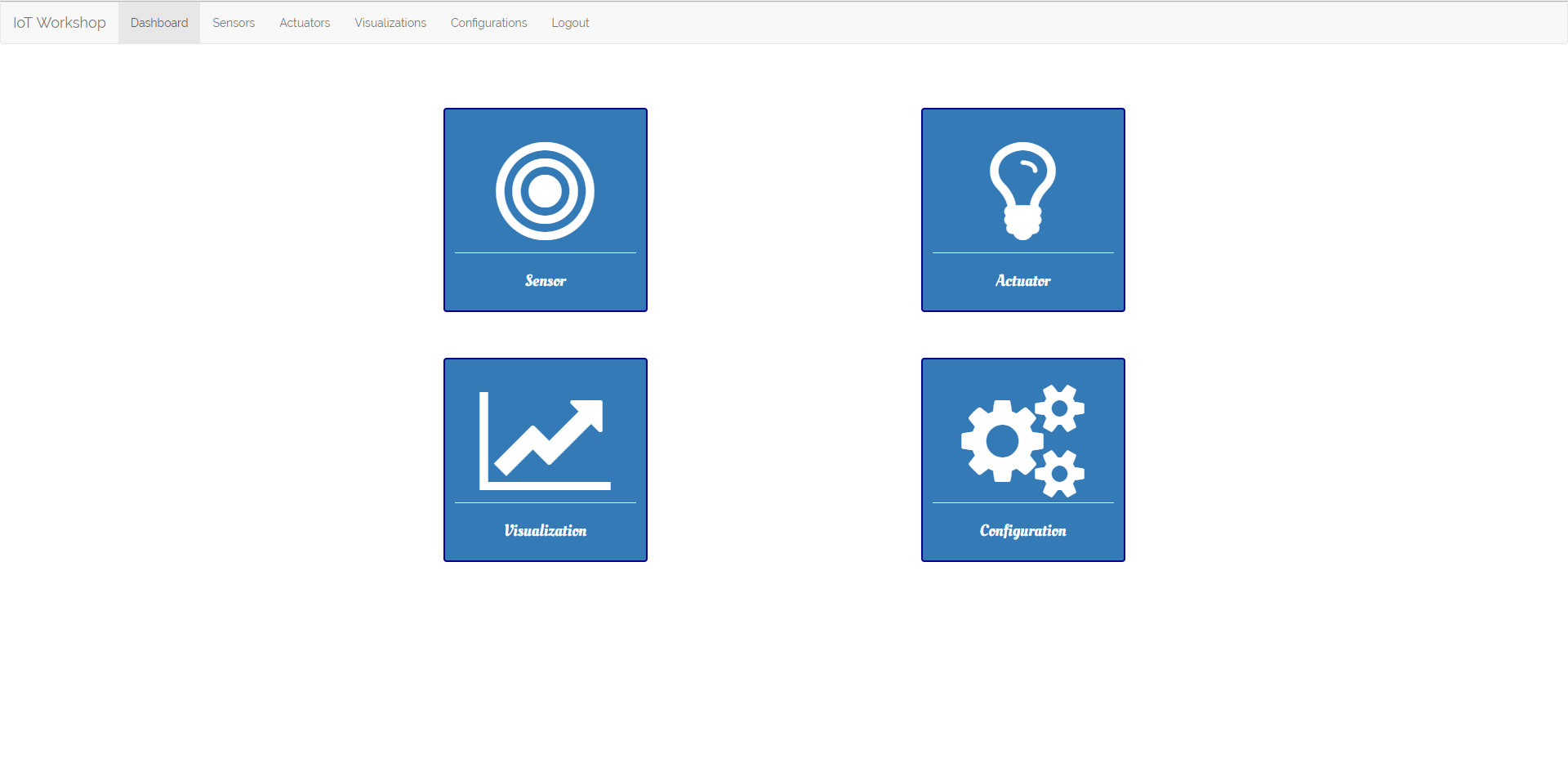
Sensor/Actuator page 

<https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/ActuatorPageSelected.php>

<https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/SensorPageSelected.php>

The sensor or actuator page is used for selecting the module that is linked to the sensor or actuator nodeMCU device. By pressing the submit button the form sends a GET request to the API. The API handles the GET request by putting the selected sensor/actuator module on active and links the nodeMCU ID to it. The selected sensor/actuator module and the ID of the selected sensor/actuator module are saved in the cookies on the client side.

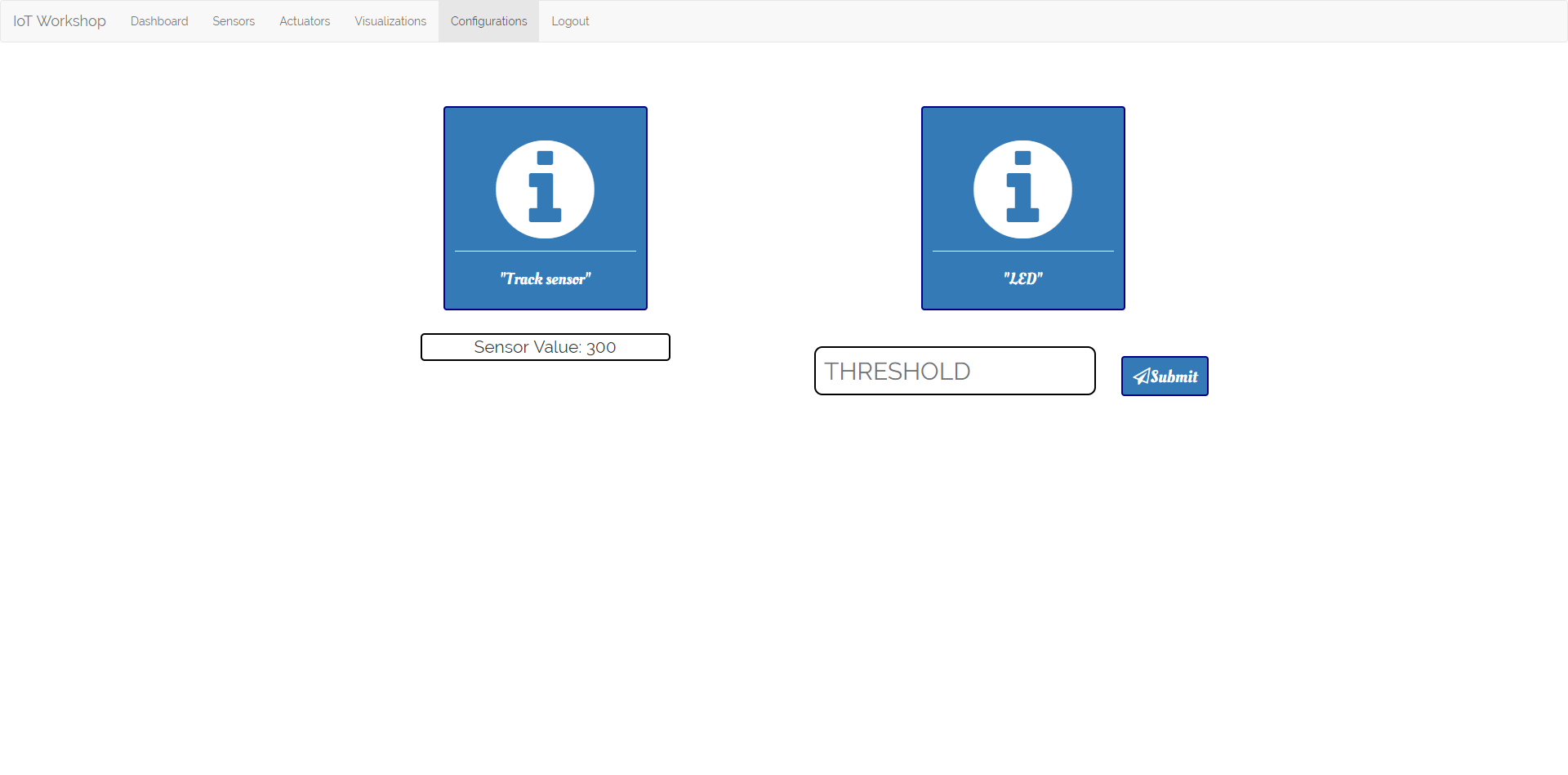
# Dashboard



<https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/Dashboard.html>

The dashboard page is developed to make the navigation of the website clear to the user. On this page, the user can view all of the function the website has and can navigate easily by using this page. The page is made of some buttons with JavaScript to it to make the buttons more user friendly. By clicking on a button the website links to the page of the button.

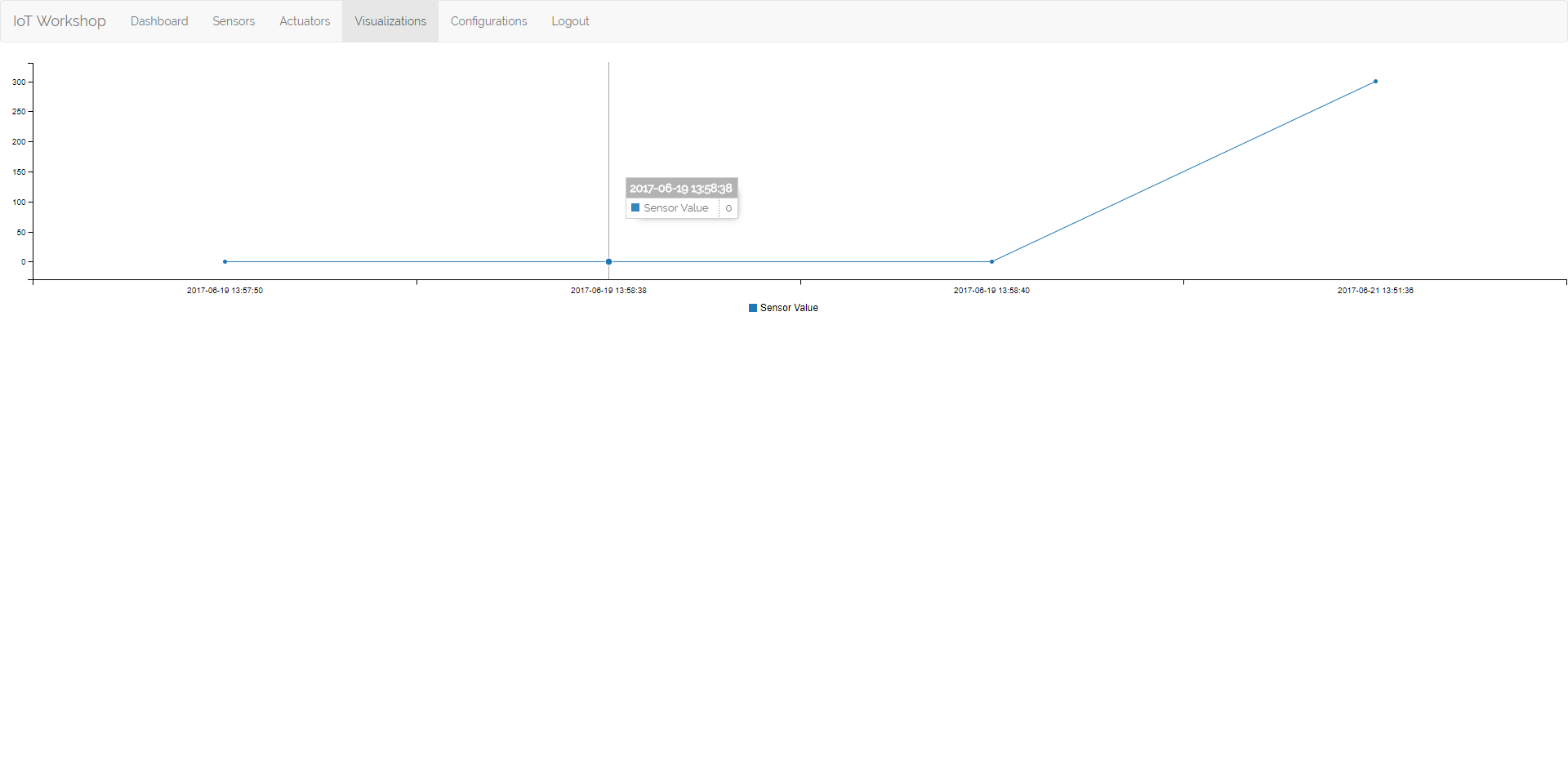
# Configuration page



<https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/ConfigurationPage.php>

The configuration page is mainly used to configure the Actuator so that it can respond to the corresponding sensor value. The configuration page can also serve extra information of the actuators and sensors by clicking on the buttons. When clicked on one of the information buttons the website opens a new tab with the wiki page of the sensor/actuator. This way the user get some more information and can think of the problems he can solve with the sensor/actuator combination. The configuration page features real time data of the sensor in text form so you can easily tune your actuator. AJAX is used to fetch data real-time from the database.

# Visualization page



<https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/VisualizationPage.php>

The visualization page is used for displaying data to the user. This data is live data and the graph updates itself every second. The visualization is built with D3.js and C3.js. Those are two libraries specifically designed to display data. With D3.js you can customize your visualization as much as you want. We tried to keep it as simple as possible so the user doesn’t get confused. D3.js is a pretty big library, to simplify it we used C3.js on top of it. This made it a lot easier to create a data visualization and made it easier to update it real-time. We used AJAX to update the graph real-time and to fetch the latest sensor data of the database. We did this by making an AJAX call to a PHP script that fetches the latest sensor data from the database. When tabbing out of the visualization page the graph might bug and zoom out automatically. This is a bug in the D3.js library and the user has to renew the page when this happens.

# Bootstrap framework

We made use of the bootstrap framework to make the website responsive for all devices. We used a navigation bar of the framework for easier navigation. The navbar “collapses” on mobile devices and shows a hamburger menu so the user can navigate through the website. We also made use of the buttons of the framework because they looked better than the default buttons.

# Code documentation

All of our code is commented so other people should be able to use it with no problem. We tried to make our code as readable as possible.

We made use of JavaScript and CSS to make our website look better.

The CSS file can be found at: <https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/style.css>

The JavaScript can be found at: <https://gitlab.dmci.hva.nl/berhanf001/project-iot-workshop-healthcare/blob/master/Website/Dashboard.js>