Assignment 2

Names: Sivert Grønli Amundsen, Mons Stenbråten
Emails: siverga@stud.ntnu.no, monsst@stud.ntnu.no
Student numbers: 539675, 539685
Link to our Heroku page:
https://assignment2-clouddev12.herokuapp.com/

Scenario: We chose to implement cloud systems in a smart boat. Here the user can see what the IOT datas is at any time. We chose to send data of : x x x, which is passed to mongodb as XML format. Here we can receive the XML data on a website, so that the user can watch the data on a webpage.

Functions: We have a few functions in our app.

```
// A function to generate a random AQI number.
function randomAQI(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

This is a function that generates a random AQI number for our smoke sensor. We use Math.random to get a random number between the max and min limit we can put as parameters later on. We used 0-500, as this is the scale when measuring AQI (level of air pollution).

```
//A function to generate a random boolean (true or false)
function randomBoolean() {
  var random_boolean = Math.random() < 0.5;
  console.log(random_boolean);
  return random_boolean;
}
```

This is a function that generates a true or false (boolean), currently since it has a <0.5, it means that it has a 50/50 chance to be either true or false.

```
// A function to generate a random temperature number.
function randomWaterTemp(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

This function generates a random water temperature between the given max and min. We gave the range 0-30, as seen below.

```
Temperature: randomWaterTemp(0, 30),
```

```
// Transforms our data from JSON to XML.

var message = json2xml(data);
client.on("connect", () => {
  setInterval(() => {
    client.publish(topic, message);
    console.log("message sent", message);
    randomBoolean();
    randomAQI();
    randomWaterTemp();
  }, 10000); //refreshes every 10th second.
});
```

The next function is an arrow function that transforms the data from JSON to XML. The function also has a given interval of 10000 which is 10 seconds, this means that it will be pushed a random generated data every 10th second to our mongoDb database.

```
broker.on('published', (packet) => {
    message = packet.payload.toString()
    console.log(message)
    mongolink.connect(url, (error, client) => {
        var myCol = client.db('IoT').collection('IoT_Data');
        myCol.insertOne({
            message: message
        }, () => {
            console.log('Data is saved to MongoDB')
            client.close()
        })
    })
})
```

The broker.js file is the component that makes it possible to work with mqtt clients. The function above, sends the data we wrote in pub.js, to the mongoDB database. In the code we chose what to call the database, and what to call the collection the data will be stored in.

Components: In our system we have 3 components.

A component to detect when the boat has a leakage.
A component to detect the water temperature.
A component to detect smoke levels on the boat.

All of the different components display what kind of sensor, and the location on the boat. Each component also displays some kind of value, like if there is any leakage, the water temperature, and the smoke level measured in AQI.

```
var data = {
  Data: {
    SOM: {
      Tab: [
        {
          Values: {
            Type: "Water Level Sensor",
            Location: "Hull",
            Leakage: randomBoolean(),
          },
        },
        {
          Values: {
            Type: "Temperature Sensor",
            Location: "Bottom of vessel",
          },
          Temperature: randomWaterTemp(0, 30),
          Unit: "Celcius",
        },
        {
          Values: {
            Type: "Smoke Sensor",
            Location: "Cockpit",
          },
          Smokelevel: randomAQI(0, 500),
          Unit: "AQI",
        },
      ],
    },
  },
};
```

Communications: Water Level Sensor, Temperature Sensor, Smoke Sensor > MQTT Broker > My Computer

Issues we faced:
As you may have noticed by now, we have had some problems when developing this app.

We have not been able to successfully upload on heroku. We have pushed code to the app, but we are just getting errors on the application when trying to open it…

Our functions which generate random data such as temperature and AQI are working in the terminal, as we are getting random values every 10th second, but when it's posting it to the mongodb database, we only get multiple copies of the first generated value. We tried to put the functions in the setInterval function, as we thought this would fix the problem, but as mentioned, it only displays random values in the terminal when posting.

We also tried to implement some type of authentication, but it caused us some errors, so therefore we commented it out on the server.js file. (line 16-73)