My Smart Home

I was able to fulfil most of the requirements for this project. The home page and about page just included one line each about the website. The add device page functioned as wanted and added the device to the database. There was also a check status page where you had to select the device from a dropdown and click the button to check its status. Similarly, update and delete pages allowed you to select a device and redirect you to another page where the action wanted can be performed.

The success feedback was shown by redirecting to another page in the module.

```
//redirects to updateDeviceStatus page to show action is successful
res.redirect("/updateDeviceStatus");
```

In the hot reloading, when the device was updated, I added the values in input boxes from the database but set it 0 for checkboxes as it gave an error when I would change the device type from a device with on/off property to an open/close one.

```
<!--Html text input for device name and intial valuse the same as database-->
<label for="device_name"> Give your device a name:</label>
<input id="device_name" type="text" name="device_name" minlength="4"  value=<%= devices.name %>  required/> <br>

<!--Html checkboxes for device on/off and open/close. Initially set to 0 instead of the previous value to avoid error
whem the device is changed from on/off to one with open/close-->
<label for="switch"> On/Off: </label> <label class="switch">
<input type="checkbox" name="onoffCheckbox" id="onoff" value=0 onChange="checkOn()"> <span class="slider round"
required></span></label> <br>

<!--Html number input for temperature and volume with min and max range and start value as the value in database-->
<label for="switch"> Open/Close: </label> <label class="switch">
<input type="checkbox" name="opencloseCheckbox" id="openclose" value=0 onChange="checkOn()"> <span class="slider round"
required></span></label> <br>

<label for="temp"> Set Temperature (Degree Celcius between 10-30): </label>
<input id="temp" type="number" name="temp" value=<%= devices.temperature %> min="10" max="30" required> <br>

<label for="vol"> Set Volume (0-100): </label>
<input id="vol" type="number" name="vol" value=<%= devices.volume %> min="0" max="100" required> <br>
```

For non-applicable fields, I disabled the appropriate fields using javascript. I also added verification in the form using min max values.

```
document.getElementById('onoff').disabled = false;
document.getElementById('openclose').disabled = true;
document.getElementById('temp').disabled = false;
document.getElementById("vol").disabled = true;
```

The values which weren't applicable were assigned -1 in the module.

```
//Adding data to database
let sqlquery = "INSERT INTO devices (device_type , name , on_off, open_close, temperature,volume) VALUES (?,?,?,?,?,?)"
// execute sql query
let newrecord = [req.body.selectDevice, req.body.device_name , -1 , req.body.opencloseCheckbox, -1, -1];
db.query(sqlquery, newrecord, (err, result) => {
if (err) {
      return console.error(err.message);
```

Appropriate get and post endpoints were used. Lastly the CSS was mostly done from scratch but some code was used from w3 schools for example the switch.

```
/*---------------------------------- Add Device Page -------------------------------
//Gets the addDevice.html page on path /addDevice
app.get("/addDevice", function (req, res) {
    res.render("addDevice.html");
});

//post for the form submission
app.post("/addingDevice", function (req,res){

    //Gets the device type selected by user
    const devicetype = req.body.selectDevice;

    //If conditions to check devices
    //Group together devices with similar values


    //Devices with on/off and temperature
    if(devicetype == 'Cooling' || devicetype == 'Heating' || devicetype == 'Fridge')
```

I initially planned on showing the status of devices on the same page when a dropdown option was selected but unable to do that, I had to add another page for show, update and delete pages. Moreover, I was unable to make the two buttons in a form perform different actions so I made a link look like a button in delete page instead.

This app is a very basic one and the designing can be improved significantly. There could be checks for names so there aren't any devices with the same values and name could be set as the primary key.

I used just one table with the name devices. It had an auto increment id set as the primary key to make each device unique. Moreover it had a varchar device name and device type. It included booleans on/off and open/close which would be set to -1 when the device doesn't have the functionality. Lastly there were integers temperature and volume which had a range of values.

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
device_type	varchar(50)	YES		NULL	
name	varchar(50)	YES		NULL	
on_off	tinyint(1)	YES		NULL	
open_close	tinyint(1)	YES		NULL	
temperature	int	YES		NULL	
volume	int	YES		NULL	

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Con

id	device_type	name	on_off	open_close	temperature	volume
31	Blender	blend	1	-1	-1	-1
32	Cooling	cooling	1	-1	18	-1
36	Heating	House Heating	1	-1	20	-1
37	Hob	hob1	1	-1	-1	-1
38	Fridge	upstairs fridge	1	-1	10	-1
39	Door	Main door	-1	1	-1	-1
40	Speakers	Living room speakers	1	-1	-1	59
41	Microwave	microwave	1	-1	-1	-1
NULL	NULL	NULL	NULL	NULL	NULL	NULL