Report

URL Link:

https://hub.labs.coursera.org:443/connect/sharedlfzysoyk?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject

R1: Home Page

R1A: Used an H1 tag on the index.html page to display the name of the website

R1B: Created a navigation bar in the header which includes all the links to the other pages using the "HREF" attribute

R2: About Page

R2A: Included the navigation bar in the About page, and also added my name as a developer after some simple instruction of the web application

R3: Add Device Page

R3A: Included the navigation bar, and also displayed a form for users to add devices into the database. The input fields include name, type, switch, timer, intensity and volume of the devices. I have also added 20 different types of appliances using a dropdown so that it is easier for the user to choose their devices.

R3B: The data keyed into the form will be collected and passed into the database. If the adding of devices is successful, a message on a new page indicating that the operation was successful will be displayed.

R3C: If the input field for the appliances that the user has chosen is not applicable, I have added a script for the input field to be automatically assigned to "0".

```
var device = document.getElementById("type").value;
if(device == "Lights" || device == "Heater" || device == "Security Camera")
{
    document.getElementById("switchInput").value = "On";
    document.getElementById("timerInput").value = "0";
    document.getElementById("intensityInput").value = "";
    document.getElementById("volumeInput").value = "0";
}
```

R3D: I added a script to validate the forms of empty input. If the fields are not all filled up, a message will show up to remind users to fill up all fields.

```
function validateForm() {
    //get the value of the input field
    let x = document.forms["device"].value;
    //setting boundaries
    if(x == null)
    {
        alert(req.body.x + ": Fields must be filled out");
        return false;
    }
}
```

R4: Show Device Status Page

R4A: Included the navigation bar, and also putting the devices in the database into a table format on HTML

R4B: The types of devices that will appear in the table have been included in the <p> tag before each table. The related setting displayed for the devices also depends on which type of device it is.

R4C: For each device, not all fields apply to them so the table only display the related fields applicable for the different devices.

For Lights, Heater and Security Camera:

ID	Name	Type	Switch	Intensity	Update	Delete
1	Dining Lights	Lights	On	2	Update	Delete
9	Bathroom Heater	Heater	On	35	Update	Delete

For Television, Speaker and Radio:

ID	Name	Type	Switch	Volume	Update	Delete
6	User TV	Television	On	3	Update	Delete
8	Bedroom TV	Television	On	3	Update	Delete

R4D: I have added style in the CSS file to change the style of the table displaying the devices. The colour of the rows for odd and even ID is also different.

For Lights, Heater and Security Camera:

ID	Name	Type	Switch	Intensity	Update	Delete
1	Dining Lights	Lights	On	2	Update	Delete
9	Bathroom Heater	Heater	On	35	Update	Delete

R5: Update Device Status Page

R5A: Included the navigation bar, and in the Show Devices page, users can also click on the update button at the side to bring them into the Update Devices page

For Lights, Heater and Security Camera:

ID	Name	Type	Switch	Intensity	Update	Delete
1	Dining Lights	Lights	On	2	Update	Delete
9	Bathroom Heater	Heater	On	35	Update	Delete

R5B: All the fields will also be displayed for users to change its setting. The only setting that they cannot change would be the ID number as it is unique to each device.

R5C: For fields that do not apply to the devices, I have added a script for the input field to be automatically assigned to “0”.

R5D: The table shown on the Show Devices page have a little update button at the side to bring users to the Update Devices page for them to change the setting of their current devices in the database.

```
//UPDATE DEVICES
app.get("/updatedevice", function (req, res) {
    res.render("updatedevice.html");
})
app.post("/deviceupdated", function (req, res) {
    //saving data in database
    let sqlquery = "UPDATE devices SET name=?, type=?, switch=?, timer=?, intensity=?, volume=? WHERE id=?"

    //execute the query
    let newrecord = [req.body.name, req.body.type, req.body.switch, req.body.timer, req.body.intensity, req.body.volume, req.
    body.id];
    db.query(sqlquery, newrecord, (err, result) => {
        if (err) {
            return console.error(err.message);
        } else {
            res.send("This device has been updated - Name: " + req.body.name);
        }
    });
});
```

R6: Delete Device Page

R6A: Included the navigation bar, and in the Show Devices page, users can also click on the delete button at the side to bring them into the Delete Devices page.

For Lights, Heater and Security Camera:

ID	Name	Type	Switch	Intensity	Update	Delete
1	Dining Lights	Lights	On	2	Update	Delete
9	Bathroom Heater	Heater	On	35	Update	Delete

R6B: A little pop-up will appear when users click on the delete button to ask for confirmation before deleting the device. If the device has been deleted successfully, a message on a new page indicating that the operation was successful will be displayed.

```
<td><a href="deletedevice" onClick="return confirm('Are you sure?')">Delete</a></td>
```

```
//DELETE DEVICES
app.get("/deletedevice", function (req, res) {
    res.render("deletedevice.html")
});
app.post("/devicedeleted", function (req, res) {
    //saving data in database
    let sqlquery = "DELETE FROM devices WHERE id=" + req.body.id;

    //execute sql query
    let newrecord = [req.body.id];
    db.query(sqlquery, newrecord, (err, result) => {
        if (err) {
            return console.error(err.message);
        } else {
            res.send("This device has been deleted - Device ID: " + req.body.id + ", " + req.body.name);
        }
    });
});
```

Database Structure

Purpose:

I have added all the different types of devices in one table for easier access.

All the different fields not applicable to the devices have been set to automatically assign it to “0”.

Field Names:

ID, Name, Type, Switch, Timer, Intensity, Volume

The switch includes on, off, open and close.

The intensity refers to speed, temperature or brightness.

id	name	type	switch	timer	intensity	volume
1	Dining Lights	Lights	On	0	2.00	0
2	Living Room Fan	Fan	On	20	2.00	0
6	User TV	Television	On	40	0.00	3
8	Bedroom TV	Television	On	20	0.00	3
9	Bathroom Heater	Heater	On	0	35.00	0
10	Kitchen Dishwasher	Dishwasher	On	30	0.00	0
11	Laundry Washing Machine	Washing Machine	On	20	0.00	0
12	Baking Oven	Convection Oven	On	50	180.00	0
13	Front Gate	Smart Gate	Open	0	0.00	0
14	Bedroom Blinds	Roller Blinds	Close	0	0.00	0
15	Seafood Freezer	Freezer	On	0	-14.00	0
16	Kitchen Refrigerator	Refrigerator	On	0	3.00	0

12 rows in set (0.02 sec)

Data Types:

I mainly used varchar for texts and int for digits.

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
type	varchar(50)	YES		NULL	
switch	varchar(5)	YES		NULL	
timer	int	YES		NULL	
intensity	decimal(5,2)	YES		NULL	
volume	int	YES		NULL	

7 rows in set (0.01 sec)