Home Automation

Application Requirement:

A JavaScript application simulating house automation: pressing a button on a control panel would visually turn on a light, change the temperature or close the curtains. Some constraints:

- the application must use jQuery.
- the components must have HTTP based "server" interaction (use a static file for simplicity, data persistence is not required). For example, the heating component retrieves the current temperature from the server and also sends the desired one back to the server.
- the solution has to be extensible and documented, so that we can develop our own components that react to events.

The application will be executed on a plain HTTP server with no possibility to run code server side and is being viewed in 2 major browsers of your choice.

The goal for creating this home automation panel is creating an automation system using static assets and visual effects. The current implementation is done in jQuery and some asynchronous calls are made for server interaction.

The layout for home is a typical home layout which is converted in to SVG for providing the visual lighting animations.

As mentioned earlier the project has been created in jQuery. The parameters are served to the server through jQuery POST request and response received from server. The process involved the following

- The image was converted to SVG and then id is assigned to different shapes.
- The DOM is accessed with jQuery.
- Different inputs are converted to “situations” and with respect to each situation different colors updated in SVG
- With every change or event, data is being posted to server and response received from the server
- jQuery library version 3.7.1 is loaded as static asset.
- All the styles are defined in the CSS file which provides the look and feel.
- The automation() is the main function which contains the logic for the Home automation and all the variables need to pass the data/ requests are made in this function.

The App can be divided in to following components:

1. **Home:** It is the main module of the application or wrapper that contains all the sub modules and also the control and temperature panel.
2. **Sub-Modules/Rooms in Home:**
    a. First Bedroom
    b. Second Bedroom
    c. Living Room
    d. Dining Room
    e. Kitchen
    f. Hall
    g. Stairs
    h. Temperature Panel
3. **Features:** When you check the options in panel, you can switch on or off lights in any of the mentioned rooms. You can also open and close curtains.

Example of switching effects in Bedroom #1:

```
if(!bedroom1_lightson && !bedroom1_curtainsopen)

{

        bedroom1_svg.setAttribute("fill","#2F50EA");

}

else  if(!bedroom1_lightson && bedroom1_curtainsopen)

{

        bedroom1_svg.setAttribute("fill","#4764ED");

}

else if(bedroom1_lightson && !bedroom1_curtainsopen)

{

        bedroom1_svg.setAttribute("fill","#637BEF");

}

else
```

```
        {
                bedroom1_svg.setAttribute("fill","#7C91F1");
        }
```

JSON Data Mapping

```
{
    "temp":temp,
    "bedroom1":{
      "lightson":bedroom1_lightson,
      "curtainclosed":!bedroom1_curtainsopen
    },
    "bedroom2":{
      "lightson":bedroom2_lightson,
      "curtainclosed":!bedroom2_curtainsopen,
      "secondcurtainclosed":!bedroom2_curtainsopen_1
    },
    "livingroom":{
      "lightson":livingroom_lightson,
      "curtainclosed":!livingroom_curtainsopen
    },
    "bathroom":{
      "lightson":bathroom_lightson,
      "curtainclosed":!bathroom_curtainsopen
    },
    "kitchen":{
      "lightson":kitchen_lightson,
      "curtainclosed":!kitchen_curtainsopen
```

```
},
"hall":{
  "lightson":hall_lightson,
  "curtainclosed":!hall_curtainsopen
},
"stairs":{
  "lightson":stairs_lightson,
  "curtainclosed":!stairs_curtainsopen
}
```


Reference:

The inspiration is taken from Home Automation by Mary Shaw:

https://github.com/marybeshaw/Just-Another-Automated-Home