ECE 448/528 – Application Software Design
Spring 2025

Project 6: Web Frontend for IoT Hub

Due: Sunday, May 4th, 2025, midnight
EXTENSION REQUEST WILL ONLY ALLOW 3-DAY EXTENSION

IMPORTANT:	***You must sign and date below acknowledgment statement below on the title page of your report.*** **Failing to do so, or any violation of this rule will result in an automatic failure for this course.**
Acknowledgment:	I acknowledge all works including figures, codes, and writings belong to me and/or persons who are referenced. I understand if any similarity in the code, comments, customized program behavior, report writings, and/or figures are found, both the helper (original work) and the requestor (duplicated/modified work) will be called for academic disciplinary action.

I. Overview

In this project, we are going to build the web frontend for our IoT hub to utilize the RESTful services provided by the server backend.

You should be able to find the user stories describing the requirements in the next section. While we are going to discuss possible class designs and implementations in the lectures, you are free to choose designs and implementations you are comfortable with. For this project, you are required to design testing procedures, one for each user story, to test the application. Since our user stories are mostly based on UI interactions, the procedures will be like instructions for users to use the application. An example procedure is provided for one of the user stories.

Once the testing procedures are documented, you are required to demonstrate that the web application works as expected. Please use screen capture software (e.g., Zoom) to record your operations together with audio explanations. Upload your video along with your report to Canvas. If the file is too large, upload your video to OneDrive and share the link within the report with access privileges to both the instructor and the TA. The details should be found in Section IV. Deliverables and Grading.

While in general, you don't need to modify the code for the server backend, you may find bugs in it when debugging your web frontend. Please correct those bugs as needed.

ECE 448/528 – Application Software Design
Spring 2025

II. User Stories

1. Plugs and PlugStates

As an end-user, I want to see available plugs and their states, so that I can know what plugs are there and whether they are on or off.

2. Control a Single Plug

As an end-user, I want to click a button on the web page to switch on/off or toggle a plug of my choice, so that I can easily control it.

3. Groups and Plugs

As an end-user, I want to see available groups, as well as plugs belonging to a group of my choice and their states, so that I can know what groups have been defined, and the state of a group.

4. Group Management

As an end-user, I want to add groups and modify their members on the web page, so that I can easily manage them.

5. Control Plugs in a Group

As an end-user, I want to click a button on a web page to switch on/off or toggle all plugs belonging to a group of my choice, so that I can easily control them together.

6. Multi-User Synchronization

As an end-user, I want to see the state update for plugs in all places if someone else switches on/off or toggles plugs from another browser so that multiple users can use the web application together.

III. Additional Requirements

To shorten your implementation time, you should not alter the UI style (CSS) already provided in the course project code (`public/index.html and associated .js`) and the lecture code (`public/members.html and associate .js`). This means that your `public/index.html` UI style should remain the same as given in the course project code. In fulfilling User Stories 3-6, you should use the same UI style provided in the lecture code. If you change the UI style, you will be given a 10-point deduction towards your Project 6 grade. For clarification, please refer to the lecture recordings.

ECE 448/528 – Application Software Design
Spring 2025

IV. Testing Procedures

Here is an example testing procedure for the user story "Plugs and PlugStates". As you can see, the procedure is organized as an ordered list of items. For each item, we first state the operation you will perform and then state the expected output on the UI. You may use it as it is, modify it, or design your own as needed.

1. Open a browser and access the root page. The names of all available plugs will show on the left part of the page.

2. Click a plug name on the left. Its state should show on the right part of the page.

3. Click a different plug name on the left. The right part of the page should show the state of that plug now.

V. Deliverables and Grading

Please push your code to the Git repository before the deadline. Although there is no auto-grading for Project 6, we will check your code to evaluate its quality and will run your code to verify your demonstrations.

Please submit a project report to Canvas before the deadline. This report should include 6 testing procedures, one each for the 6 user stories. You may copy my example procedure for the first user story into the report if you choose to use it. Please submit the demonstration video to Blackboard together with the report.

The 100-point project grade will be evaluated using the following criteria:

- Source code quality (40 points):
    - Properly formatted source code (10 points): with indentation and reasonable line width.
    - Reasonable implementation (10 points): meaningful names and purposes for JavaScript variables, methods, and classes.
    - Reasonable user interface design (10 points): proper use of HTML/CSS features.
    - Reasonable RESTful communications with backend (10 points).
- Testing procedures (30 points): 5 points for each testing procedure that correctly addresses one user story.
- Demonstrations (30 points): 5 points for each successful demonstration of one testing procedure that can be verified by the TA using the code you have pushed to the Git repository.