

Rapid Prototyping and Experimentation 2  
Independent Study Module CS-IS-4031-1

**Prof. Debayan Gupta**

**With Sai Khurana**

Pranav Iyengar

Spring 2023

---

## **LED Graph**

### **Objective**

The objective of this project was to create LED graph hardware that could be controlled by the students of Ashoka University, This graph was to have 3 parts - a hardware component, a web client for users and a middleware component to connect the hardware and web client

The repository containing the code for the web client -  
<https://github.com/Makerspace-Ashoka/led-graph-mono>

### **My Role**

My role in this project was to create a web client that would enable the users to manage and create custom configurations that would display different animations on the hardware. The configurations should be visualized on the web client.

### **Milestones achieved**

The different features I integrated into the project are as follows:

1. Typescript express backend to store user and config data
2. Front end client
3. Firebase backend and storage (DEPRECATED)

The features of the backend are:

1. Route to handle login through google authentication. It only allows users with ashoka IDs to login
2. Route to view all of a users configurations
3. Routes to upload and delete configurations
4. Route to view a configuration in the visualizer

5. Route to transfer configuration from backend to middleware through sockets (for faster transfer)
6. Integration with Prisma and MySQL to handle data

The features of the frontend are:

1. An About page detailing the project and how to use the web client
2. A content management system for a user's configuration data
3. User login and logout
4. A visualizer that emulates the physical hardware, allowing users to see their configurations before creating new ones
5. An interactive JSON editor that allows configs to be edited (DEPRECATED)
6. Screening of vulgar words when creating a config

## Set Backs

There were several setbacks I faced when creating the web client. These are:

1. Time spent waiting for definition of hardware positions and design. Until I knew what the hardware would look like, I could not emulate it
2. Time spent waiting for the definition of the protocol and config schema. Until I knew how the data had to be formatted and what kind of data the middleware needed, I could not move forward with critical sections of the software

## Pitfalls

The pitfalls of the project are as follows:

1. Dockerization of the software is incomplete as there was a setback with containerizing the Prisma instance
2. The software does not have a validation section for admins to decide which configurations are allowed to be sent. All configurations are currently valid
3. The client and the middleware are not currently in constant communication. The code for the transfer is written into the web client, and I have made a dummy program that accepts the data, but the middleware is not currently accepting it (***not under my scope of work***)
4. The middleware does not have the same LED mappings as the web client. For example, the led which is mapped to position 1 in the web client is not the same led which the middleware maps to 1 on the physical hardware. This mapping should be done by the middleware (***beyond my scope of work***)

## Result

The result of this project is a web client that allows users to sign up and upload configurations that they can visualize in a software emulated version of the LED Graph hardware. Currently, the configurations are not being received by the middleware, but that is beyond the scope given to me. Users have complete control over each LED, and they can create configurations that work at a refresh rate of 10 frames per second. This visualizer works with the current physical hardware of 3 hexagons, and **can easily be scaled to many more hexagons.**

The web client has been well documented and made scalable so that it can be either

1. Open Sourced for contributions
2. Continued to be developed by me
3. Handed over to another developer with a short amount of time taken for familiarization with the codebase.

As per the scope of work given to me, I have completed a visualizer that allows users to view their uploaded configuration files. The transfer protocol is completed on my end, and the code has been well documented and made scalable for future development.

This was my first time working with hardware, and it taught me a lot - different aspects of programming that are different from code that would be a part of a SaaS platform or a standalone project.