Open Lab

Instructional Designer University Center for Teaching and learning

Group 3 members:
Yun He(yuh95@pitt.edu)
Ruoyu Huang(ruh17@pitt.edu)
Yixiao Li (yil210@pitt.edu)
Tao Tao(tat73@pitt.edu)
Cai-Cian Song(cas386@pitt.edu)

Table of contents

1. **Background**

Our clients are planning to embedded a real-time status of open lab for general public. When students or the public who have interested in open-lab, they can know whether the staff is in Open lab now and which machine is working. After several discussions. Also, our clients want to build up a mobile application or a website for the public and staffs in Open lab. Based on this requirement we made an agreement to extend the functionalities of their proposal, and provide the features on the new website:

1. Login System for staffs only
2. Show ongoing task & machines' status information
3. Contact us
4. Showcase
5. Video Streaming
6. Content management

2. **User analysis**

1. Our users are our clients, staffs of Open lab, and students.

2. For our clients, they want us to provide features to help them to register attendance, and then embed a real-time status of Open lab for general public. They want to be able to contact the students online. They want us develop functions to allow them to do live video streaming. In addition, they think it's nice to provide a way for them to manage all the data.

For students or the public, they want to schedule appointment with open lab by this new system and take a look of the products of Open lab.

3. But in the clients; point of view, features are not the most important thing. Easy usability is more important. They hope the system can work on the mobile phone, which may make them comfortable.

On students' side, they want to use browser on laptop to access the system. This is the most common way they use.

4.And for clients, they don't care whether the interface is beautiful or not. Usability and features are what they care most. However, the students want to visit cool websites. Boring interface may cause their rejection to use the system.

3. **Task analysis**

a. <u>Login Function:</u>

Staffs can login by clicking "Staff Login" hyperlink. They are required to input their username and password. The log in function will

verify user information by comparing the user data on Google Airtable. If the input match the data on Airtable, then the staff can login the system and have authority to modify pages.

After logging in the system, some of the functions and pages will change. In index page, the “Subscribe” button will change to “Update” button and has different function. User can use “Subscribe” to send their subscription to staffs while Staffs can use “Update” to push information on website.

If the input does not match, the server will return a different response code and the function can catch the code and output relative error message. If there is an unexpected error, the function will also show an error message with different content.

b. Send new events to subscribe:

Users can send subscribe to staffs. This function requires users input their email address and choose one or more content they are interested in. An email will be automatically generated by users’ input and send to one of the staffs’ email address through background program. Then the subscribe content will be added to Airtable as a record in a relative table. If this function fails to connect to Airtable, it will return an error message.

c. Video streaming:

Staffs can start a video streaming on Youtube and the streaming video will be shown on our website automatically. This feature provide a new opportunity for Open lab to expose to the public and Pitt students who are interested in Open lab. Let user to know more about the facilities in Open lab.

d. Embedded real-time status in Open lab website:

Once a staff starts to work, the staff can update the machines’ working status and his/her working hours. When the data have been inserted into airtable. The embedded view on teaching center will get the latest log and render on the teaching center website.

e. Connected data from Airtable:

The connection to Airtable is easy to build up; however, for the security sake, we need to insert a PHP layer between front-end and Airtable. More detail can be found in section 6 implementation.

f. Content management pages:

Our clients use Airtable as their database to store data. And the most content in their account is not relevant to our new website. Thus, our clients want us to create our own data. Once we finish the new website,

they can simply update api_key to connect to their data. And our content management can help them fix a problem which is to inform Teaching center website staffs to upload new data in the database of Teaching center. Because the information on Teaching center is uploaded by staffs in Teaching center, and their current working process is communicated via email. When there is a new information from Open lab, our clients need to email Teaching center to ask them to help to upload a new announcement. In our website, when our clients create a new data in event table, the back-end will automatically send an email to inform Teaching center staffs.

g. Switch mode:

Once staffs login our website, the homepage provides a switch button to redirect to the data management pages and vice versa. This function allows staffs to switch pages from front-end pages to back-end pages.

h. Showcase viewer:

This feature allows users to browse previous works. Once users click the picture in the showcase list, the picture will be amplified in a modal view and there will be author name and description in the view.

i. Punch-in/punch-out:

To allow a staff to update their working hours, and a staff can announce that which machine is working. Furthermore, every record will be saved in Airtable for future references.

4. User study

a. The user study for prototype 1

We ran a user study based on our prototype 1, we designed the questions below:

1. Ask what kind of contact function they prefer.
2. Ask whether they like video streaming and showcase.
3. Ask whether they need to add more functions.
4. Ask whether they like our user interface design.
5. Ask whether the experience is good enough when they use our site.

The feedback from our first user study are listed below:

1. Subscribe the website to receive the latest information because staffs are not always there.
2. Students suggest we can only keep email contact.

3. The theme color can be brighter.

b. The user study for prototype 2

This time our material included the website and the embedded view on teaching center. This time we conducted the user study on our clients, students, and some residents in Pittsburgh.

1. Ask whether they like video streaming and showcase.
2. Ask whether they need to add more functions.
3. Ask whether they like our user interface design.
4. Ask whether the experience is good enough when they use our site.

The feedbacks from students are pretty similar to our first result. Most students can use our website easily and they usually intent to click the pictures in the showcase tab to amplify the previous work. More feedback are listed below:

- Feedbacks from general public:

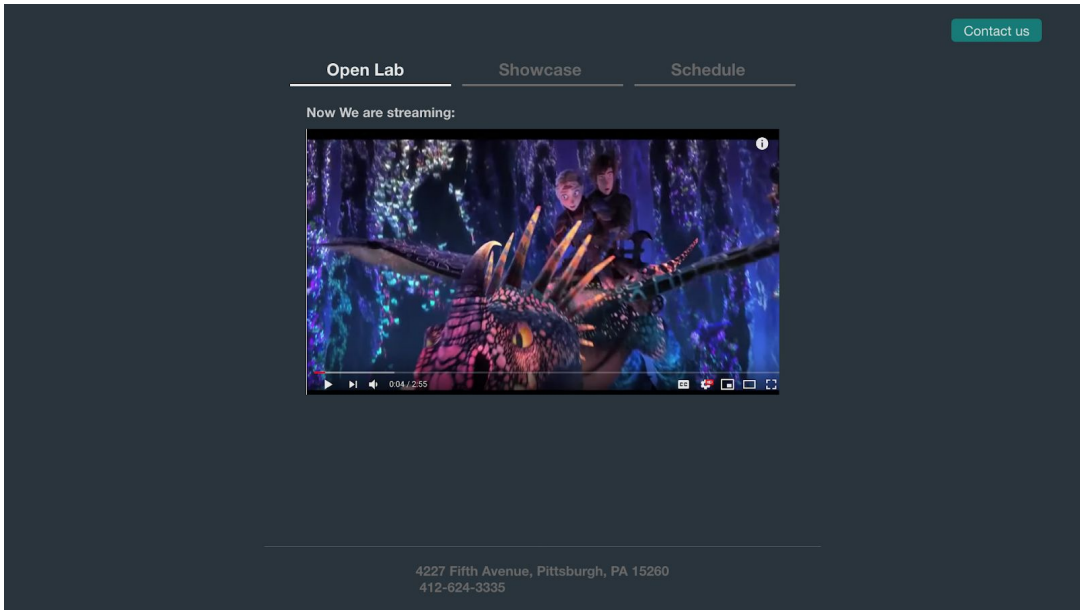
1. The website should provide more detail about the next video streaming event. In order to let the information of Open lab can be exposed to the public, there should be a way to broadcast the relevant information and upcoming events.
2. It is not user-friendly to implement Contact us function with outlook API. Because when users click the Contact us button, the system will prepare a pop up window, which is build-in mail system in any operating system. And it will bother some users to use this function because some of them are no used to use build-in mail system.

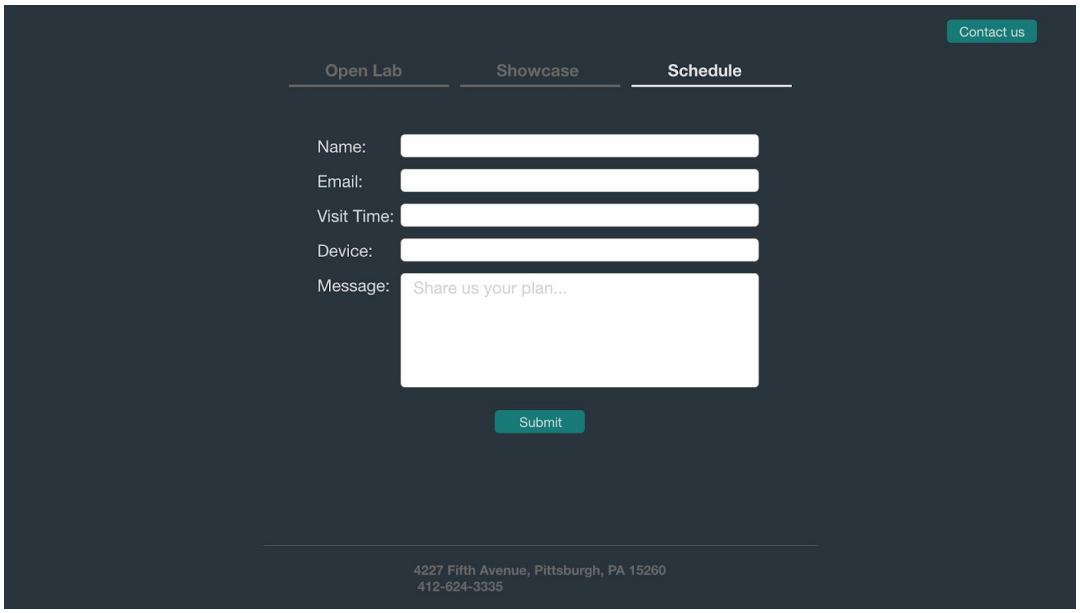
- Feedbacks from our clients:

1. The design of embedded view need to be updated because the Teaching center website is going to updated. So the original design is not suitable for new style.
2. Our website is going to run on Teaching center site, so the staff from Teaching center wants us to make the style can be looked the same as one of Teaching center site.

5. User interface design

a. Version 1

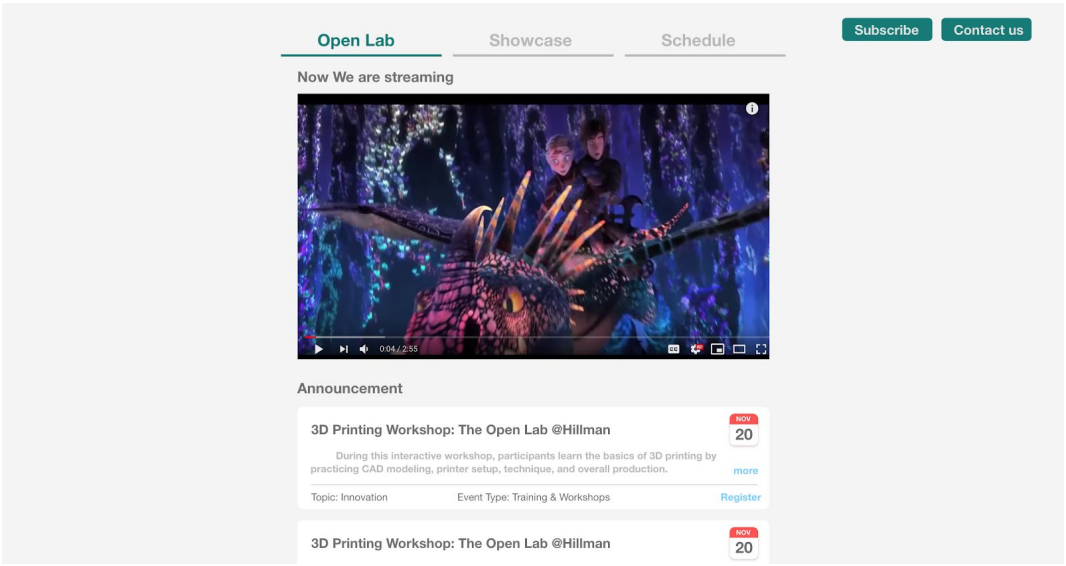


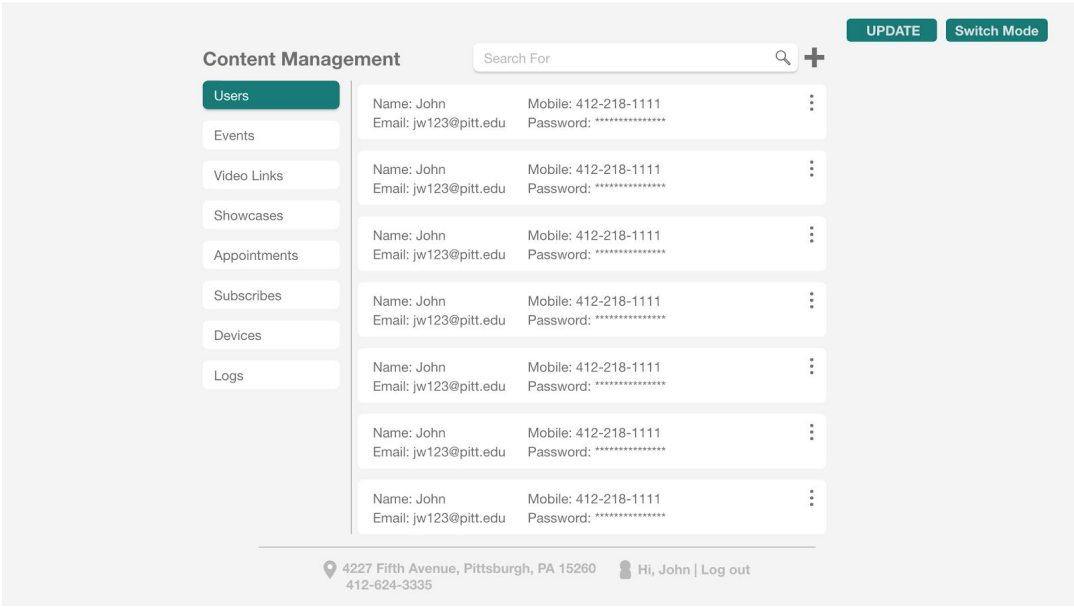


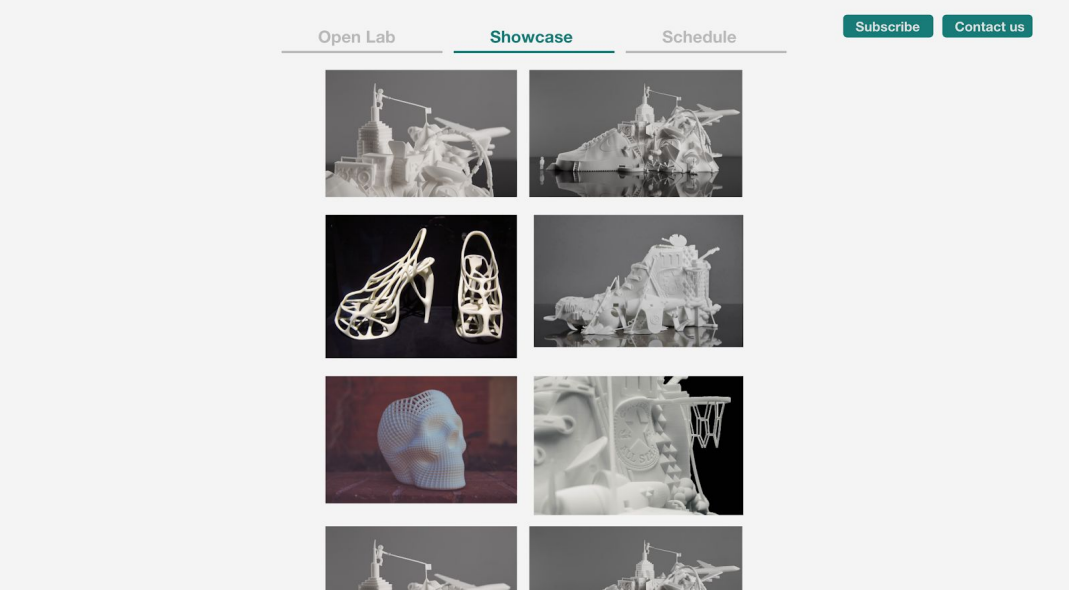
b. Version 2

After we collected user feedback from our first user study, users highly recommended us to change the theme color lighter, and it would be nice to allow users to receive the latest information from Open lab. According to these two main feedback from the first , we updated our design and planned to add a new feature which allows users to subscribe Open lab, and users can receive announcements when a staff post a new event.

The pictures below is our version 2:

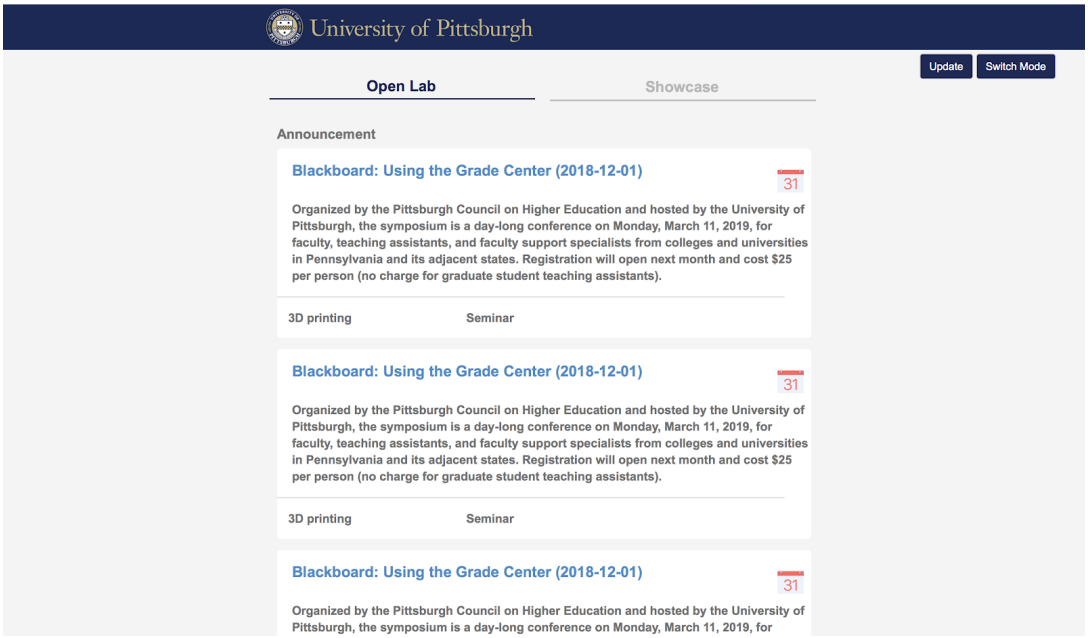

Open Lab
Showcase
Schedule
Subscribe
Contact us
Now We are streaming
0:04 / 2:55
Announcement
3D Printing Workshop: The Open Lab @Hillman
NOV
20
During this interactive workshop, participants learn the basics of 3D printing by practicing CAD modeling, printer setup, technique, and overall production.
more
Topic: Innovation
Event Type: Training & Workshops
Register
3D Printing Workshop: The Open Lab @Hillman
NOV
20

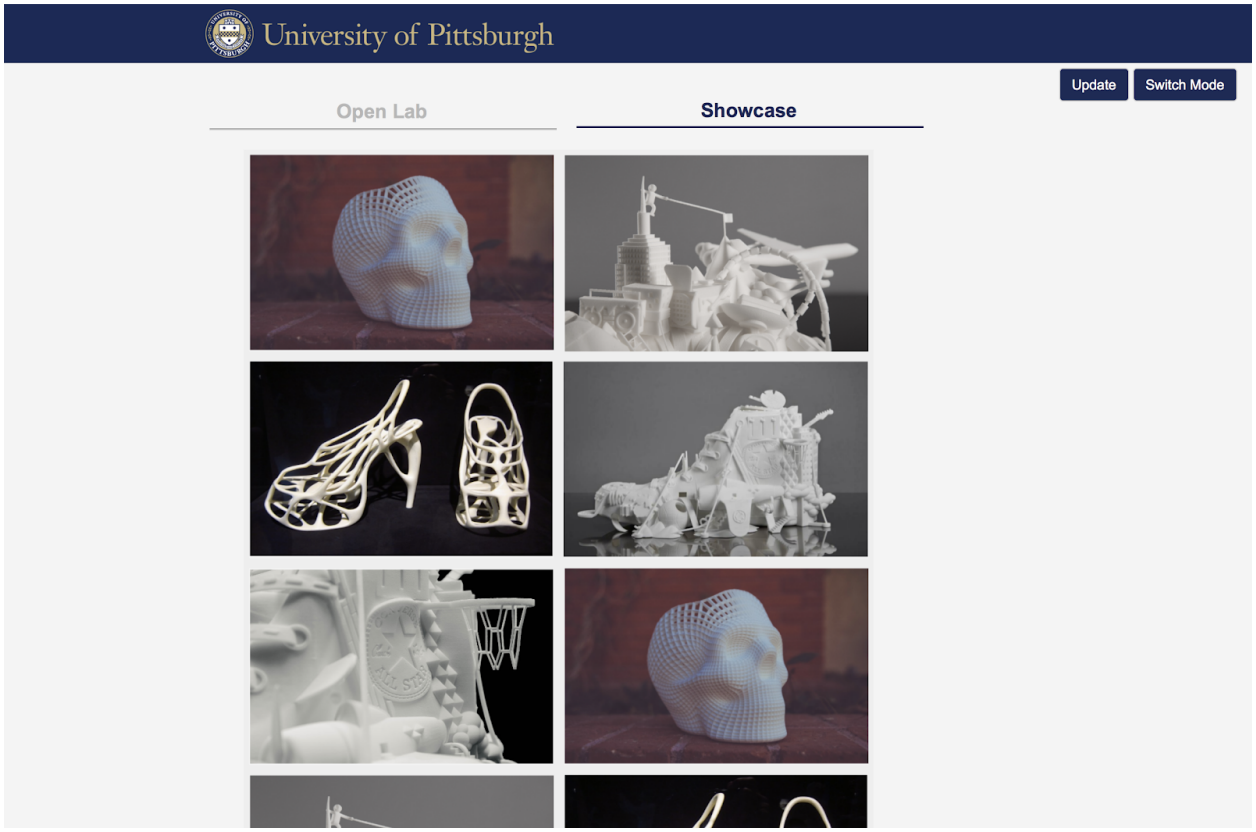

UPDATE
Switch Mode
Content Management
Search For
Users
Events
Video Links
Showcases
Appointments
Subscribes
Devices
Logs
Name: John
Email: jw123@pitt.edu
Mobile: 412-218-1111
Password: **********
4227 Fifth Avenue, Pittsburgh, PA 15260
412-624-3335
Hi, John | Log out



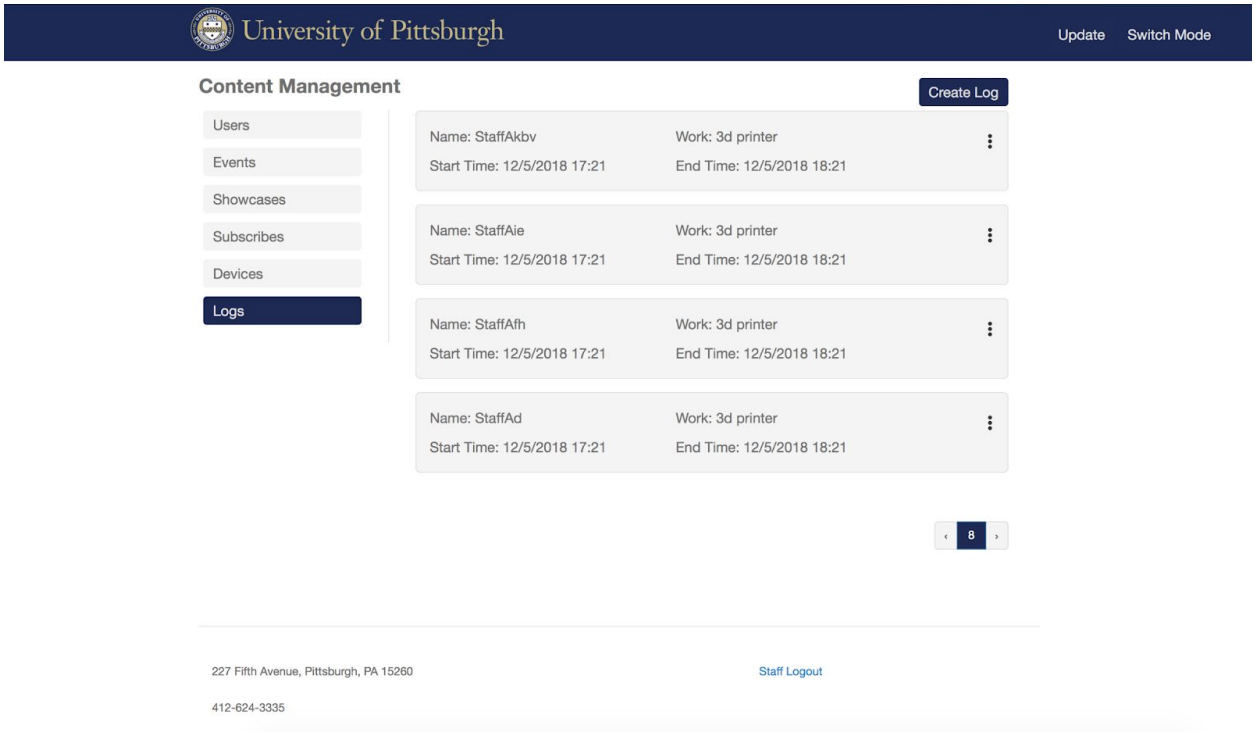
c. Version 3

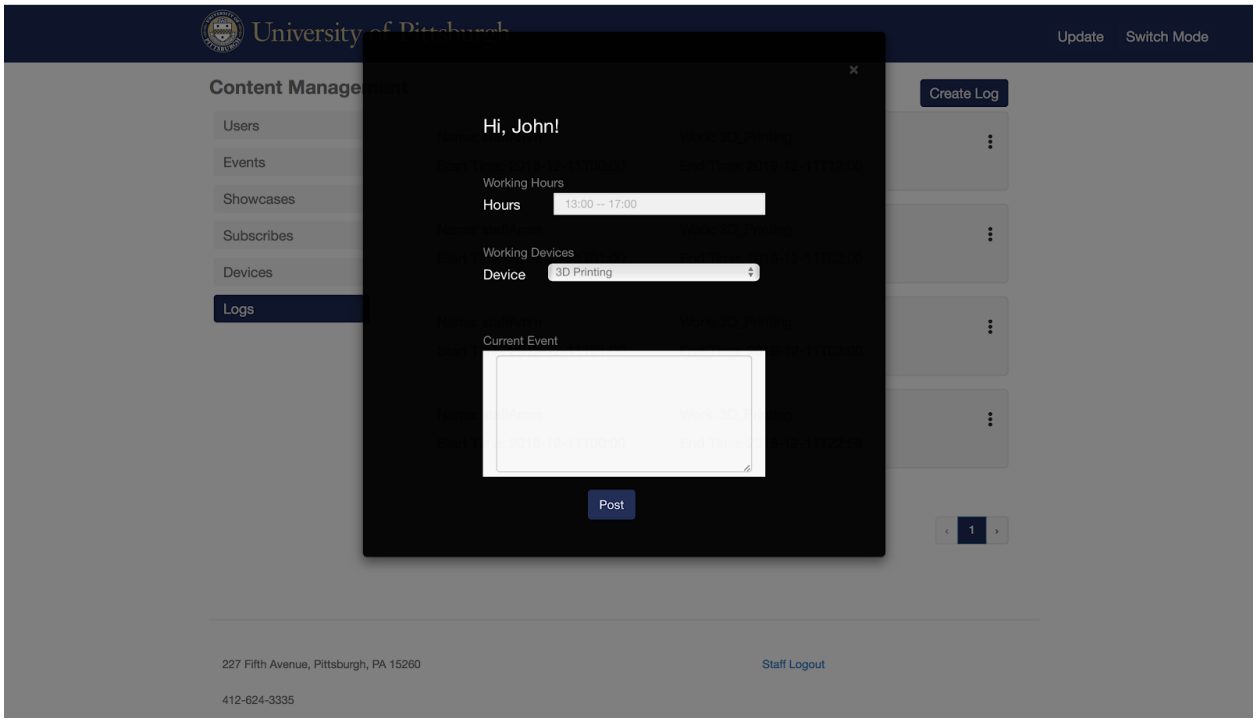
After we had a meeting with Teaching center staffs, they decided to run our website on their current site as a sub folder. And staff from teaching center want us to update the style. Based on this reason, we have to update our style again. The theme should be similar with the Teaching center style. Here is our final design:

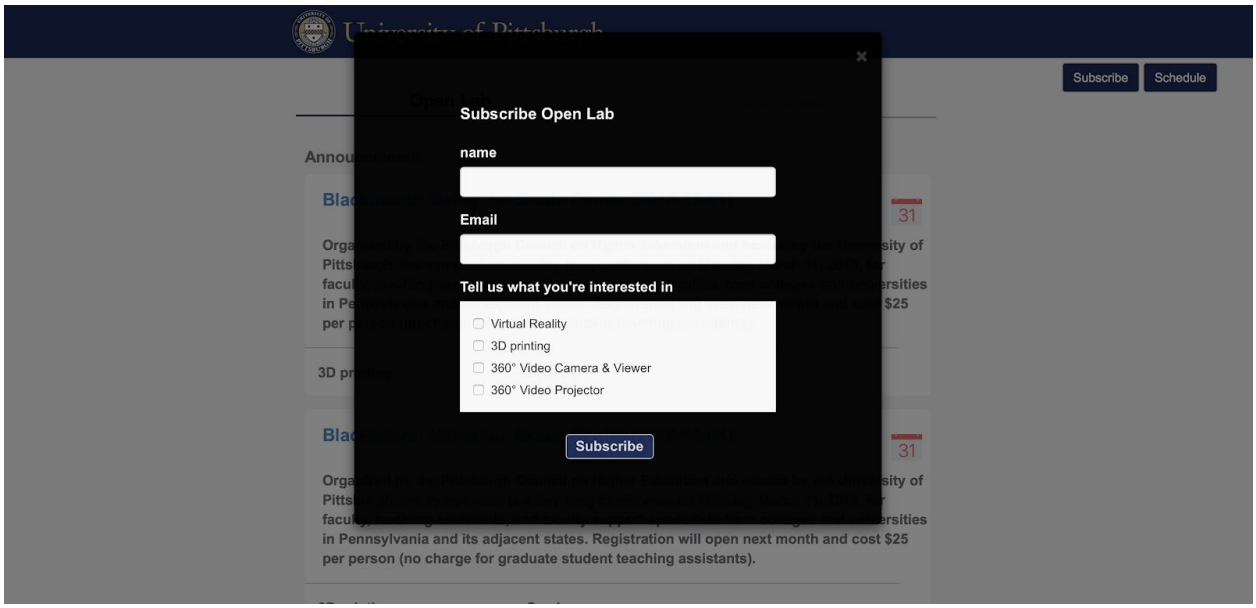



University of Pittsburgh
Update
Switch Mode
Open Lab
Showcase


University of Pittsburgh
Update
Switch Mode
Open Lab
Showcase
Great shoe Author: Jerry Time: 09/11/18
Image 4 of 10

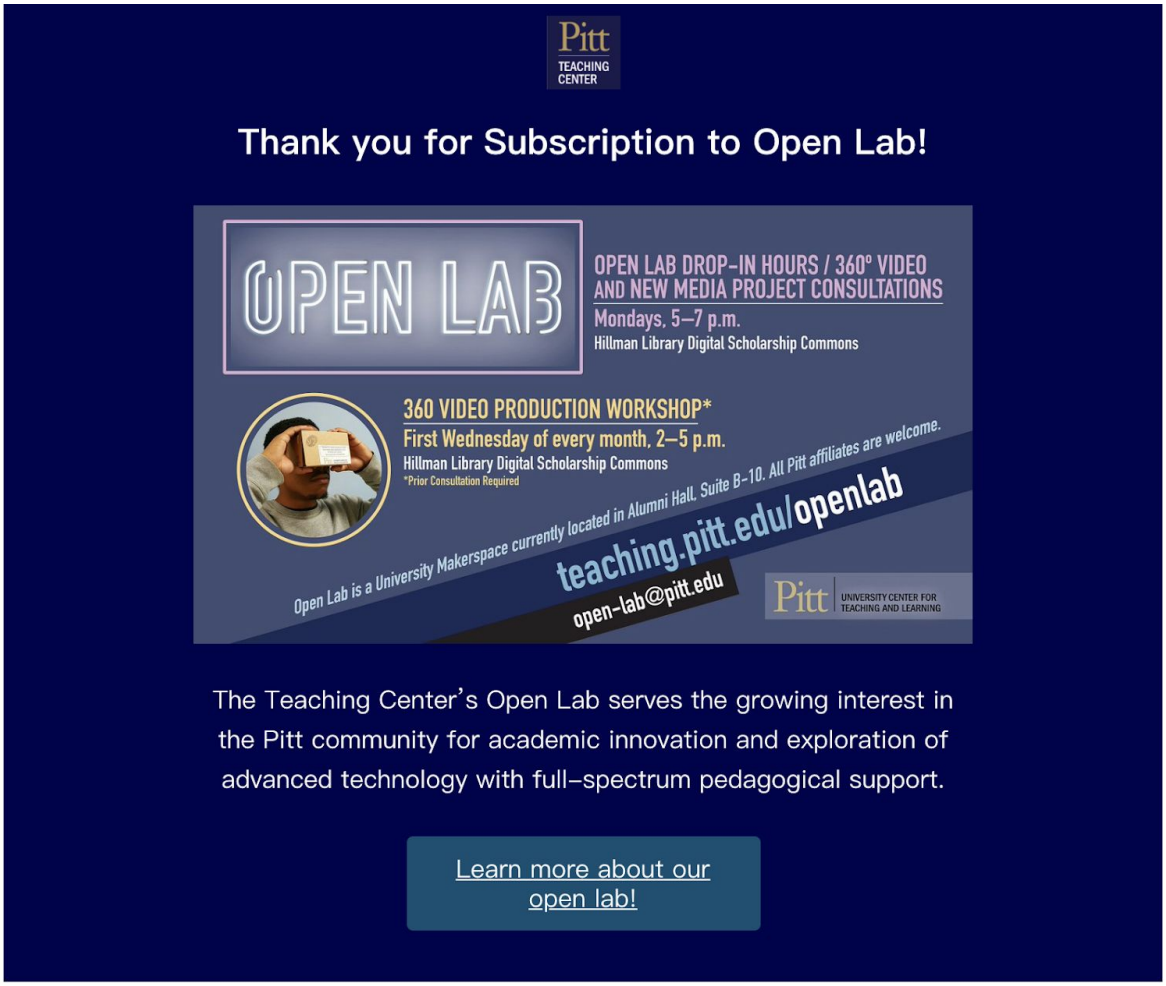

University of Pittsburgh
Update
Switch Mode
Content Management
Create Log
Users
Events
Showcases
Subscribes
Devices
Logs
Name: StaffAkbv
Work: 3d printer
Start Time: 12/5/2018 17:21
End Time: 12/5/2018 18:21
Name: StaffAie
Work: 3d printer
Start Time: 12/5/2018 17:21
End Time: 12/5/2018 18:21
Name: StaffAfh
Work: 3d printer
Start Time: 12/5/2018 17:21
End Time: 12/5/2018 18:21
Name: StaffAd
Work: 3d printer
Start Time: 12/5/2018 17:21
End Time: 12/5/2018 18:21
8
227 Fifth Avenue, Pittsburgh, PA 15260
412-624-3335
Staff Logout


Update
Switch Mode
Create Log
Users
Events
Showcases
Subscribes
Devices
Logs
Hi, John!
Working Hours
Hours
13:00 -- 17:00
Working Devices
Device
3D Printing
Current Event
Post
1
227 Fifth Avenue, Pittsburgh, PA 15260
412-624-3335
Staff Logout



Subscribe

Email content 1



Email content 2

Lear more about Showcase1

The perfect choice for any purpose of a message.

Learn more about Showcase2

The one with an action button seen without scrolling.

Learn more about Showcase 3

The best for introducing a feature or inviting to an action.

Learn more about Showcase 4

Great one for offering a variety of options to explore.

What's more...

Contact:412-666-6666 Email:openlab@pitt.edu
You can change subscription settings anytime.

6. **Implementation**

Back-end:

For security issue and the extended functionalities, we decided to implement a middleware via PHP, wrapping every APIs up. When our clients make some operations based on creating new data into Airtable as a trigger, back-end can process other functions simultaneously. And we also implement a encrypt session mechanism to make sure that once our clients' session expired, the account can be log out automatically. The pictures below are part of our internal API document.

How to use:

URL = baseurl + table name + fields(or parameters) + sess

- **baseurl:** localhost/isd_final/connection/index.php
- **fields(optional, provided by airtable):**
  - **fields(string array):** to filter return data with certain column
  - **maxRecord(number):** The maximum total number of records that will be returned in your requests. If this value is larger than pageSize (which is 100 by default), you may have to load multiple pages to reach this total. See the Pagination section below for more.
  - **pageSize(number):** The number of records returned in each request. Must be less than or equal to 100. Default is 100. See the Pagination section below for more.
  - **offset(string):** If there are more records, the response will contain an offset. To fetch the next page of records, include offset in the next request's parameters.
- **sess(must):** Return from server when user login.

Example:

-- Get user list(method: GET): http://localhost/isd_final/connection/index.php/users

-- Get one specific user data(method: GET): http://localhost/isd_final/connection/index.php/users/rec0eMTQiJn7JxDHn

-- POST a new date(method: POST): http://localhost/isd_final/connection/index.php/users/ parameters in body with json format:

Example:

-- Get user list(method: GET): http://localhost/isd_final/connection/index.php/users

-- Get one specific user data(method: GET): http://localhost/isd_final/connection/index.php/users/rec0eMTQiJn7JxDHn

-- POST a new date(method: POST): http://localhost/isd_final/connection/index.php/users/ parameters in body with json format:

```
{
    "username": "Eden",
    "account": "cas386@pitt.edu",
    "password": "1234",
    "user_phone": "122312312"
}
```

-- UPDATE data(method: PUT): http://localhost/isd_final/connection/index.php/users/rec0eMTQiJn7JxDHn parameters in body with json format:

```
{
    "username": "Adam"
    "user_phone": "122312312"
}
```

-- DELETE data(method: DELETE): http://localhost/isd_final/connection/index.php/users/rec0eMTQiJn7JxDHn

---

Response:

- Request succeeded:

```
{
    "code":200
    "data": {
        "sess": "wjfiowqjfiojfioqw1232",
        "username": "Eden"
    }
}
```

- Request failed:

```
{
    "code":404
    "msg": "Something wrong!"
}
```

Front-end:

We use the front-end library of bootstrap. You can see a header logo of PITT at the head of each pages. In addition, we use flexbox grid to build layouts of all shapes and sizes, which made our system adaptive to all kinds of devices, such as smartphones, ipads and laptops. At the end of each page, it's a footer part which shows some information and provide an interface for the staffs to log in.

At the home page, at the top is the navbar which can navigate us to two different parts, open lan and showcase. At the openlab part of the index page, we offer an iframe to show video streaming. Under the video streaming is the announcement part. And at the showcase part of the index page, we put several pictures of showcases on it. What's more, we can amplify each picture which enable the users to see more details of the showcase. I think it's a good design.

The users of this system is not only students who want to see video streaming and samples, but also the staffs of open lab. So we have a content management system for staffs.

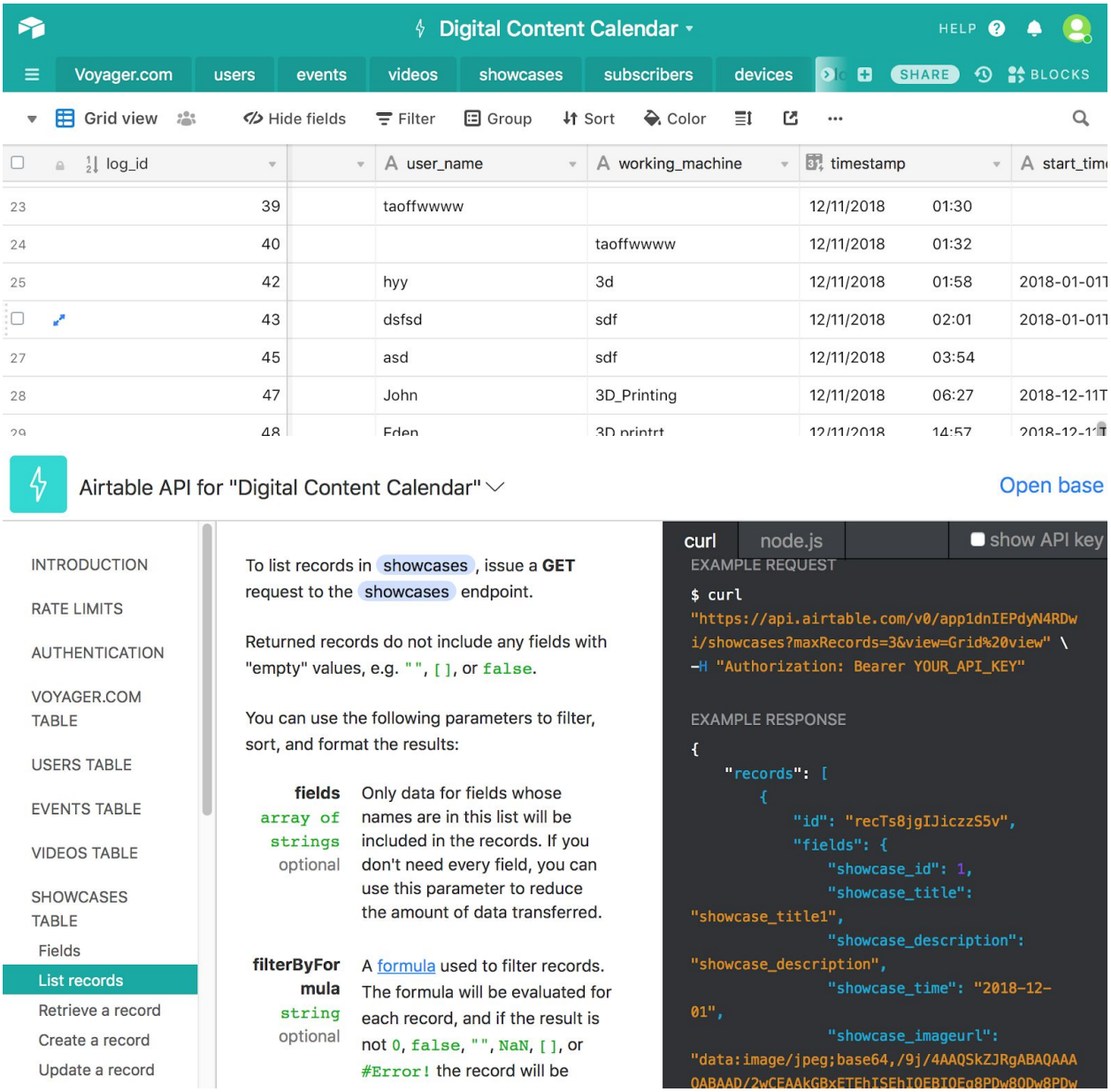
On the left side of content management system are navigation lists which can navigate to each parts of the content management system. On the right side, staffs can see all the records of logs, events, and etc. Because of the large numbers of records, we provide an pagination function to show the records. Also, staffs can delete and edit each records. If it is necessary, they can add new record in this system.

Third-party API:

1. Airtable

   We implemented a session mechanism, which encrypt/decrypt users' information. Users need to pass their session to server, and the back-end will decrypt the session into username, last login time. Once the session is expired, the back-end will return error, and the front-end gets the response with error codes. The system will log out automatically.

   The implementation of the part is to create a based model to wrap up every query API, using curl to connect to Airtable. And the Airtable api_key and other configuration settings will be protected into another directory.



2. Youtube

The YouTube Live Streaming API lets you create, update, and manage live events on YouTube. Using the API, you can schedule events (broadcasts) and associate them with video streams, which represent the actual broadcast content.

The Live Streaming API is actually comprised of components of the YouTube Data API and the YouTube Content ID API. The Data API enables YouTube users to manage their YouTube accounts, while the YouTube Content ID API enables interactions with YouTube's rights management system. However, all of the resources that make up the Live Streaming API are used only to create and manage live events.

1. **Request an access token**

> **Note:** Requests to Google's authorization server must use **`https`** instead of **`http`** because the server is only accessible over SSL (HTTPs) and refuses HTTP connections.

When a user first tries to perform an action that requires API authentication, you need to direct the user to Google's authorization server at `https://accounts.google.com/o/oauth2/auth`. The table below identifies the request parameters that you need to (or can) include in the URL. Note that the request URI that you construct must contain properly URL-escaped parameter values.

Parameters	
`client_id`	**Required**. The OAuth 2.0 client ID for your application. You can find this value in the Developers Console.
`redirect_uri`	**Required**. A registered **`redirect_uri`** for your client ID. Register valid redirect URIs for your application in the Developers Console.
`response_type`	**Required**. Determines whether the Google OAuth 2.0 endpoint returns an authorization code. Set the parameter's value to **`code`**.
`scope`	**Required**. A space-delimited list of scopes that identify the resources that your application could access on the user's behalf. These values determine which permissions are listed on the consent page that Google displays to the user. The YouTube Data API supports the following scopes: Scopes

Assuming the user has granted access to your application, exchange the authorization code obtained in step 3 for a refresh token and access token. To do so, send a `POST` request to `https://accounts.google.com/o/oauth2/token` that includes the following key-value pairs in the request body:

Key-value pairs	
`code`	The authorization code that Google returned to your **`redirect_uri`** in step 3.
`client_id`	The OAuth 2.0 client ID for your application. This value is displayed in the Google APIs console.
`client_secret`	The client secret associated with your client ID. This value is displayed in the Google APIs console.
`redirect_uri`	A registered **`redirect_uri`** for your client ID.
`grant_type`	Set this value to **`authorization_code`**.

Calling the YouTube Data API

After obtaining an access token for a user, your application can use that token to submit authorized API requests on that user's behalf. The API supports two ways to specify an access token:

1. Specify the access token as the value of the `Authorization: Bearer` HTTP request header. This is the recommended approach.

   ```
   GET /youtube/v3/channels?part=id&mine=true HTTP/1.1
   Host: www.googleapis.com
   Authorization: Bearer ACCESS_TOKEN
   ...
   ```

   You can test this using cURL with the following command:

   ```
   curl -H "Authorization: Bearer ACCESS_TOKEN" https://www.googleapis.com/youtube/v3/channels?part
   ```

2. Specify the access token as the value of the `access_token` query parameter:

   ```
   https://www.googleapis.com/youtube/v3/channels?part=id&mine=true&access_token=ACCESS_TOKEN
   ```

   You can test this using cURL with the following command:

   ```
   curl https://www.googleapis.com/youtube/v3/channels?part=id&mine=true&access_token=ACCESS_TOKEN
   ```

7. Video demo

Url: https://pitt.box.com/s/m2f7q5hxnl2xt30utq38ko7kcj368pj3

8. Contribution

Yun He: Front-end layout
Yixiao Li: Front-end layout, login & admin dynamic page display logic, lightroom
Tao Tao: API connection
Ruoyu Huang: Video streaming, email content, and content in Airtable
Cai-Cian Song: Contact clients, design the website, implementing back-end, email function in back-end