Meeting Note 1: First Meeting: Oct, 25th

What did you work on?
(Finished Assignment 1)

We built the Hello Word application for Assignment 1 and analyze Youtube APIs
What will you do next? Decisions and plans?

Decisions for Frontend design:

The initial frontend design was that all the information is shown on the same page. The home page should contain a search bar, a table for channel info, a table for playlist info, and a table for video info. However, after we discussed and thought about how to create a user-friendly product with a better user interaction flow, we changed our frontend design to be separating the information into several pages. First, after the user enters a channel ID, the page should be automatically redirected to the channel info page if a channel is found by ID. Then, users can check all the information about this channel. Second, after clicking the playlist or video list button. The website will redirect to the corresponding webpages to check all information about playlists or videos in detail. There will be titles, description, videos number of each playlist or view/like/comment count of each video. In order to prepare for the final assignment, we also build 2 buttons on the channel information page for recommendation. But since these functions are in the next assignment, we have not finished them yet. While browsing each page of our website, users can search any time on the navigation bar and click the back button to return to the previous page.

Feature/Task Prioritization:

The most important part of the current milestone is that the application should allow the user to search for a channel by channel ID. Hence, the very first feature that needs to be implemented is the navigation bar and the search box. If we don't get the channel ID from the user, it is hard to implement other features based on the user input. At the same time, Austin can work on the backend APIs since the backend APIs can be tested using postman. It does not need to wait for the search box to be finished. After the search box and navigation bar are done, we can start working on the other pages, such as the channel info, playlist info, and video info pages.

Plans:

1. Frontend
    a. Austin:
        i. Decide to build a navigation bar and pages about Youtube channel/videos (Deadline: Nov 2nd)
    b. Zixiao:
        i. Decide to build search box and page about Youtube playlists (Deadline: Nov 2nd)
2. Backend
    a. Austin:
        i. Decide to build 3 APIs for Youtube channel/playlists/videos (Deadline: Nov 2nd)
3. Test:
    a. Austin:
        i. Decide to build tests for Youtube channel/videos (Deadline: Nov 5th)
    b. Zixiao:
        i. Decide to build tests for Youtube playlists (Deadline: Nov 5th)

Meeting Note 2: Second Meeting: Nov, 2rd

What did you work on? Past work?

1. Frontend
    a. Austin:
        i. Finished implementing the navigation bar
    b. Zixiao:
        i. Finished implementing a search box, filter box, and page about Youtube playlists
2. Backend
    a. Austin:
        i. Finished the 3 APIs for Youtube channel/playlists/videos
3. Documentation: Zixiao (Started working, but not finished yet)
4. Record Meeting Note: Austin

What will you do next? Decisions and plans?

Frontend Coding Structure:

Since Zixiao was the one that worked on the info page first, she wrote the playlist info page as a React function and imported the function to the home page for data transmission and page redirection. The initial frontend coding structure design was that after the home page gets the user input from the search bar, it calls all 3 backend APIs to get the channel, playlist, and video information just once and sends it to each component when the redirection happens. However, since Austin is more familiar with React class components instead of function components, all the pages would be written as class components. Also, in order to separate the responsibility, the home page would not need to call all 3 backend APIs after getting the user input. The only responsibility of the home page is getting the user input and doing the redirection. After the page has been redirected to the info page, the info page would need to call the corresponding backend API to get the information.

Frontend UI Design:

Since there would be too much information shown on the same page if a channel has published many playlists or videos, we decide to add a “show more” and “show less” button at the end of the playlist and video info tables. Also, in the original design, there was a dropdown menu for sorting the information at the top of the playlist and video info tables. However, we found that sorting the information is not in our original design. In order to save more time for other features, we decided to remove the dropdown menu.

Plans:

1. Frontend
    a. Austin:
        i. Remove and optimize the logic of some buttons since some buttons of the navigation bar do not seem to work well in UI logic. (deadline: Nov, 5th)
        ii. Use React-Router to create multiple pages (deadline: Nov, 5th)
        iii. Build pages about Youtube channel/videos (deadline: Nov, 5th)
    b. Zixiao:
        i. Remove the dropdown menu (deadline: Nov, 5th)
        ii. Connect Youtube playlist page to multiple pages router structure (deadline: Nov, 5th)
2. **Backend**
    a. Austin:
        i. Obtain more data from Youtube videos API since the original API only gets basic video information and cannot obtain data such as view count, like count and comment count. These data are important for the next milestones/assignment(deadline: Nov, 5th)
3. **Test:**
    a. Austin:
        i. Build tests for Youtube channel/videos(deadline: Nov, 5th)
    b. Zixiao:
        i. Build tests for Youtube playlists(deadline: Nov, 5th)

Meeting Note 3: Third Meeting: Nov, 5th

What did you work on?

1. Frontend
    a. Austin:
        i. Finished optimizing the logic of some buttons
        ii. Used React-Router for redirecting to multiple pages
        iii. Finished implementing the pages for Youtube channel/videos
        iv. Added UI logic button for redirecting to other pages
    b. Zixiao:
        i. Removed the dropdown menu
        ii. Connected Youtube playlist page to multiple routing pages structure
2. Backend
    a. Austin:
        i. Finished optimizing Youtube videos API
3. Test:
    a. Austin:
        i. Finished implementing the tests for Youtube channel/videos
    b. Zixiao:
        i. Finished implementing the tests for Youtube playlists
4. Summarize Meeting Note: Austin

What will you do next? Decisions and plans?
Wrap-up for A2:
Zixiao still needs to add more details and updates in the documentation.
Both need to double-check if there's no issue with the code in the main branch and make sure that all the documents including the meeting notes are included.

Prepare for the final milestones/assignment:
(The deadline for each subtask will be set based on our calendar in Nov)

1. Frontend
    Austin & Zixiao:
        i. Design algorithms to find recommended keywords for the recommendation function
2. Backend
    a. Austin:
        i. Continuously optimize API for error checking
3. Test:
    a. Austin:
        i. Optimize tests for Youtube channel/videos
    b. Zixiao:
        i. Optimize tests for Youtube playlists