Content Library

Internal working of Content library

Asset list:

- Loading the list will fetch 50 assets per API. And the results are rendered using react-window (Fixed Grid) setup. Also have used react-window-infinite-loader for achieving the infinite scroll behaviour.
- On scrolling down, API request for the next page will be made and pagination is handled using page count.

Asset card

- Video playback is implemented using react-player as it provides callback methods for building custom controls. Have implemented play/pause, remaining time for the video.
- As the video assets added here are to be used in ad generation and uploaded to social media, I have assumed the videos will be of the aspect ratio 1:1.
- If we need to support other aspect ratio in future, the video playback can be moved inside a separate modal for providing extensive controls on playback.
- Delete option will remove the asset from the list. Rename option is left unimplemented for now.
- Asset tag is implemented to have ID associated with it, so that it will be possible to filter assets by the tag on clicking.

Other UI handlings

- Filter By and search functionalities are provided using local search. The search/filter option can be moved to API and pagination can be handled by the API.

- Debounce option can be added to search when making API call for search.
- Have used React material UI for the menu lists, initial loading related handlings.

Re-architecting to support large list of assets

- For showing a large size image, we can show the compressed version in thumbnail. On clicking the image, it can be opened in full screen mode to show the full resolution image. Images can be converted to webP format for reducing the image size.
- As we are using react-window, it shouldn't a problem with maintaining a large list of assets.
- Images and videos can be served using CDN to reduce delay.
- Linear loading option can be implemented and shown when the infinite scroller is loading next page of assets.

Integration test

- Initial rendering should fetch and list assets. This can be test by mocking the API.
- Scrolling should make an API call for fetching the next page of request.
- Once the list reaches the end, duplicate calls for API call shouldn't be made.
- Filter and search option should modify the asset list properly.
- Deleting an asset should remove it from the list.
- Similarly, rename option can be tested once implemented.

Unit test

- Search component

    - Debounce option behaviour(once implemented)
    - Entering search text should trigger the onChange action
- Radio button
    - Can be tested by passing list of options
    - Clicking an option should trigger the onChange action
- Video Player
    - Test Play/Pause functionality behaviour
    - Test "Remaining time" computation when video is playing
    - Once the video has stopped playing, the remaining time should be hidden from UI
- Menu item (Used in asset card)
    - Menu open and close behaviour
    - Clicking outside should close the menu without triggering any option