

# Documentation

## Front End Web

01 NOVEMBER 2021

Prepared by: Eufraim Lawrence Ponce

## Contents

<b>My Aim .....</b>	<b>4</b>
<b>Techs.....</b>	<b>4</b>
<b>Tools .....</b>	<b>4</b>
<b>Tasks .....</b>	<b>4</b>
<b>Trending Widgets.....</b>	<b>4</b>
Components.....	4
TrendingWidget .....	4
Issue(s) .....	4
Parameters.....	5
Source code.....	7
CSS.....	8
TrendingWidgetItem .....	9
Issue(s) .....	9
Parameters.....	9
Source code.....	11
CSS.....	12
<b>CreatePost .....</b>	<b>13</b>
Components.....	13
Custom Hooks .....	13
CreatePost.....	13
Issue(s) .....	13
Parameters.....	14
Source code.....	15
CSS.....	16
<b>CreatePostFieldText .....</b>	<b>17</b>
Issue(s) .....	17
Parameters.....	17
Source code.....	17
CSS.....	18
<b>CreatePostSelectPrivacy.....</b>	<b>19</b>
Issue(s) .....	19
Parameters.....	19
Source code.....	19

CSS.....	20
<b>CreatePostUserAvatar.....</b>	<b>21</b>
Issue(s) .....	21
Parameters.....	21
Source code.....	21
CSS.....	21
<b>CreatePostUserName.....</b>	<b>22</b>
Issue(s) .....	22
Parameters.....	22
Source code.....	22
CSS.....	22
<b>CreatePostBodyOptions.....</b>	<b>23</b>
Issue(s) .....	23
Parameters.....	23
Source code.....	23
CSS.....	23
<b>CreatePostImages.....</b>	<b>24</b>
Issue(s) .....	24
Parameters.....	24
Source code.....	25
CSS.....	26
<b>FileSelect.....</b>	<b>27</b>
Issue(s) .....	27
Parameters.....	27
Source code.....	27
CSS.....	28
<b>useFileSelect.....</b>	<b>29</b>
Issue(s) .....	29
Parameters.....	29
Source code.....	30
<b>useCreatePost.....</b>	<b>31</b>
ComponentName.....	31
Issue(s) .....	31
Parameters.....	31
Source code.....	32

## My Aim

Separate UI components and logic components.

- **Logic** => customHooks, with name with useComponent
- **UI** => JSX elements only, with name with Component

Use BEM for CSS naming.

Apply Atomic UI design. Splitting components into smaller components for reusability.

## Techs

- ReactJS
- HTML
- CSS
- Javascript

## Tools

- VSCode

## Tasks

- Add/update news trending widgets
- Update swap post to send post to back end server.

## Trending Widgets

Added Trending Widgets to fetch trending news via API ([newsapi.org](https://newsapi.org)). Currently using free account. Free accounts are not allowed to use in production or if the request is done from a browser. So, for now data is a dummy response.

## Components

TrendingWidget

TrendingWidgetItem

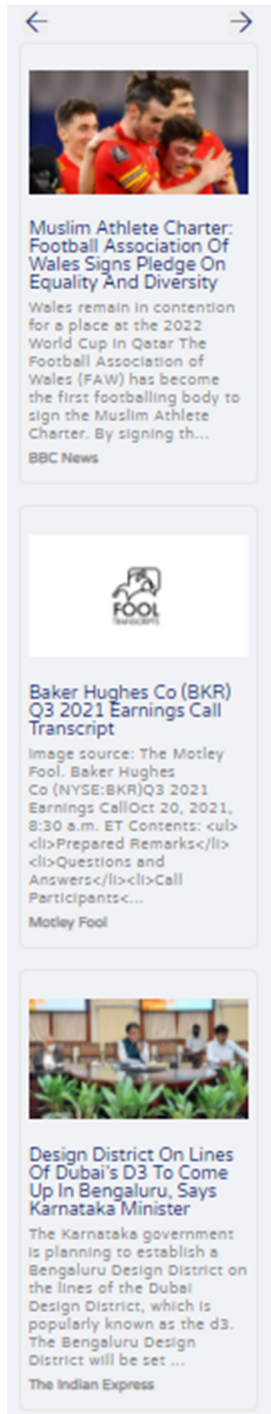
### *TrendingWidget*

The container for the trending widgets item.

### Issue(s)

1. Find an alternative API to fetch news from browser using free account.
2. Needs some Refactoring.

## Screenshot(s)



## Parameters

Name	Type	Description
pageSize	number, default is 3	How many news items to display per page.

isAutoPlay	boolean, default is true	<p>If true, TrendingWidget will automatically navigate to next page every <b>autoPlayInterval</b>.</p> <p>If false, you have to manually navigate with the nav buttons to check next page.</p>
autoPlayInterval	Number, default is 10000ms	<p>Milliseconds before navigating to next page if <b>isAutoPlay</b> is true.</p>
isDummyData	boolean, default is true	<p>If true, the response are not real.</p> <p>If false, the response will be from <a href="https://newsapi.org">newsapi.org</a>.</p>

## Source code

\*\*\*Double click to view full page.

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import './TrendingWidget.css'; //From Lawrence for Whats Trending styling
import TrendingWidgetItem from './TrendingWidgetItem';

const delay = (ms) => new Promise((resolve) => setTimeout(() => resolve(), ms));

const getDummyNewsApiResponse = async (msDelay = 1500) => {
  const { status, totalResults, articles } = DUMMY_NEWS_API_RESPONSE;
  await delay(msDelay);

  return { data: { status, totalResults, articles } };
};

const newsApiKey1 = '63dba4402f7e429aa0c9e818baf2e314';
const newsApiKey2 = '78d61a4ab2fc4b5c92149f7c22e8acbe';

const NEWS_ACTIONS = {
  PREV: 'PREV',
  NEXT: 'NEXT',
};

const TrendingWidget = ({ pageSize = 3, isAutoPlay = true, autoPlayInterval = 10000, isDummyData = true }) => {
  const [pageCurrent, setPageCurrent] = useState(1);
  const [pageMax, setPageMax] = useState(0);
  const [trendingItems, setTrendingItems] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [newsError, setNewsError] = useState(null);
  const [autoPlayIntervalId, setAutoPlayIntervalId] = useState(null);
  const [isButtonPrevDisabled, setIsButtonPrevDisabled] = useState(true);
  const [isButtonNextDisabled, setIsButtonNextDisabled] = useState(true);

  const getNews = async () => {
    try {
      setNewsError(null);
      setIsLoading(true);

      const todayDate = new Date();
      const todayString = `${todayDate.getFullYear()}-${todayDate.getMonth() + 1}-${todayDate.getDate()}`;
      const yesterdayString = `${todayDate.getFullYear()}-${todayDate.getMonth() + 1}-${todayDate.getDate() - 1}`;

      let newsResult;

      if (isDummyData) {
        newsResult = await getDummyNewsApiResponse();
      } else {
        newsResult = await axios.get(
```

**\*\*\*Double click to view full page.**

```
.container_trending {  
  width: 100%;  
  display: flex;  
  flex-flow: column nowrap;  
  justify-content: center;  
  overflow: hidden;  
}  
  
.ti-announcement {  
  height: fit-content;  
  transform: translateY(8px);  
  color: #0b2984;  
}  
  
.container_trending_nav {  
  width: 100%;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}  
  
.container_trending_nav > button {  
  border: 0;  
  color: #0b2984;  
  font-size: 24px;  
}  
  
.container_trending_nav > button:disabled {  
  color: #0b29843a;  
}
```



### TrendingWidgetItem

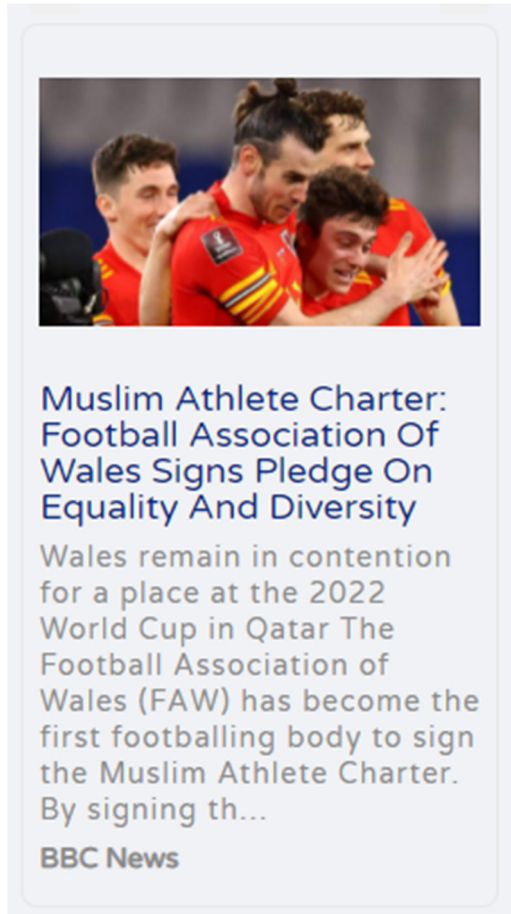
The container for the trending widgets item.

#### Issue(s)

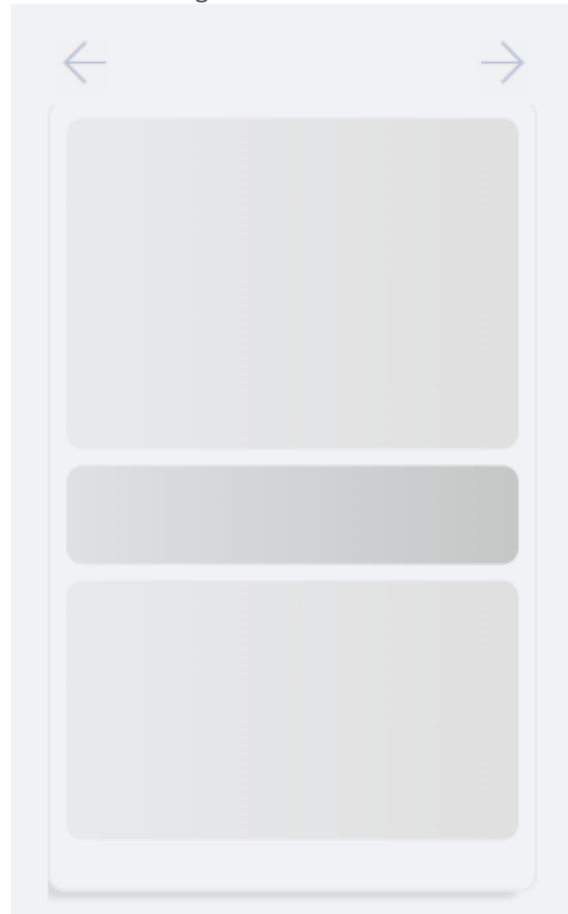
1. None.

#### Screenshot(s)

**State:** Loaded



**State:** Loading



#### Parameters

Name	Type	Description
Id	Number	Just an array index number
url	String	The redirect url when a user clicked the trending item.
urlImage	String	The url for the image source to be displayed.

title	String	The article's title.
Description	String	The article's short description
sourceName	String	The article's source name.
Loading	Boolean, default is false	If true it will display a loading UI.

## Source code

\*\*\*Double click to view full page.

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import './TrendingWidget.css'; //From Lawrence for Whats Trending styling
import TrendingWidgetItem from './TrendingWidgetItem';

const delay = (ms) => new Promise((resolve) => setTimeout(() => resolve(), ms));

const getDummyNewsApiResponse = async (msDelay = 1500) => {
  const { status, totalResults, articles } = DUMMY_NEWS_API_RESPONSE;
  await delay(msDelay);

  return { data: { status, totalResults, articles } };
};

const newsApiKey1 = '63dba4402f7e429aa0c9e818baf2e314';
const newsApiKey2 = '78d61a4ab2fc4b5c92149f7c22e8acbe';

const NEWS_ACTIONS = {
  PREV: 'PREV',
  NEXT: 'NEXT',
};

const TrendingWidget = ({ pageSize = 3, isAutoPlay = true, autoPlayInterval = 10000, isDummyData = true }) => {
  const [pageCurrent, setPageCurrent] = useState(1);
  const [pageMax, setPageMax] = useState(0);
  const [trendingItems, setTrendingItems] = useState([]);
  const [isLoading, setIsLoading] = useState(false);
  const [newsError, setNewsError] = useState(null);
  const [autoPlayIntervalId, setAutoPlayIntervalId] = useState(null);
  const [isButtonPrevDisabled, setIsButtonPrevDisabled] = useState(true);
  const [isButtonNextDisabled, setIsButtonNextDisabled] = useState(true);

  const getNews = async () => {
    try {
      setNewsError(null);
      setIsLoading(true);

      const todayDate = new Date();
      const todayString = `${todayDate.getFullYear()}-${todayDate.getMonth() + 1}-${todayDate.getDate()}`;
      const yesterdayString = `${todayDate.getFullYear()}-${todayDate.getMonth() + 1}-${todayDate.getDate() - 1}`;

      let newsResult;

      if (isDummyData) {
        newsResult = await getDummyNewsApiResponse();
      } else {
        newsResult = await axios.get(
```

**\*\*\*Double click to view full page.**

```
.container_trending {  
  width: 100%;  
  display: flex;  
  flex-flow: column nowrap;  
  justify-content: center;  
  overflow: hidden;  
}  
  
.ti-announcement {  
  height: fit-content;  
  transform: translateY(8px);  
  color: #0b2984;  
}  
  
.container_trending_nav {  
  width: 100%;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}  
  
.container_trending_nav > button {  
  border: 0;  
  color: #0b2984;  
  font-size: 24px;  
}  
  
.container_trending_nav > button:disabled {  
  color: #0b29843a;  
}
```

## CreatePost

Added Trending Widgets to fetch trending news via API ([newsapi.org](https://newsapi.org)). Currently using free account. Free accounts are not allowed to use in production or if the request is done from a browser. So, for now data is a dummy response.

### Components

CreatePost  
CreatePostFieldText  
CreatePostSelectPrivacy  
CreatePostUserAvatar  
CreatePostUserName  
CreatePostBodyOptions  
CreatePostImages  
FileSelect

### Custom Hooks

useFileSelect  
useCreatePost

### *CreatePost*

The main components for CreatePost. This will be used for normal and swap posts.

### Issue(s)

1. CSS styling need to be more updated.
2. Displaying as child element.

Screenshot(s)



Parameters

Name	Type	Description
title	String	Title to display in form.
type	String, Swap or Normal.	If Swap post will be posted as swap.  If Normal post will be posted as normal post.

## Source code

\*\*\*Double click to view full page.

```
import fileStorage from '../..//configs/fileStorage';

import useFileSelect from './useFileSelect.js';
import useCreatePost from './useCreatePost.js';

import FileSelect from './FileSelect.jsx';
import CreatePostFieldText from './CreatePostFieldText.jsx';
import CreatePostSelectPrivacy from './CreatePostSelectPrivacy.jsx';
import CreatePostUserAvatar from './CreatePostUserAvatar.jsx';
import CreatePostUserName from './CreatePostUserName.jsx';
import CreatePostBodyOptions from './CreatePostBodyOptions.jsx';
import CreatePostImages from './CreatePostImages';

import './CreatePost.css';
import { useEffect } from 'react';

// UPDATED UI HERE
const CreatePost = ({ title = 'Create Swap Post', type = 'Swap' }) => {
  const { user, post, postResponse } = useCreatePost();
  const { fileSelected } = useFileSelect();

  useEffect(() => {
    console.log('post response update', postResponse);

    if (postResponse.ok && !postResponse.loading) {
      fileSelected.fileDeleteAll();
      post.postReset();
    }
  }, [postResponse]);

  const handlePostChange = (event) => {
    const { name, value } = event.target;
    post.postUpdate({ name, value });
  };

  const handleUploadPost = (event) => {
    event.preventDefault();
    const files = fileSelected.files.map((file) => file.file);
    post.postUpload(files);
  };

  const handleFileSelect = (event) => {
    fileSelected.filesAdd(event.target.files);
  };

  const PostButton = () => (
    <button className='post-submit-swap' type='submit' onClick={handleUploadPost}>
      Share Swap
    </button>
  );
};
```

\*\*\*Double click to view full page.

```
.popup-content {
  display: flex !important;
  align-items: center !important;
  justify-content: center !important;
  flex-flow: column nowrap !important;
  padding: 0 !important;
}

.container__create-post {
  display: flex;
  align-items: center;
  justify-content: center;
  flex-flow: column nowrap;

  height: fit-content;
  box-sizing: border-box;
  width: 100% !important;
}

.create-post_title {
  font-size: 16px;
  font-weight: bold;
  font-family: 'Varela Round', sans-serif;
  box-sizing: border-box;
  padding: 16px;
  width: 100%;
  height: 100%;
  text-align: center;
  border-bottom: 1.5px solid #b89f9f;
}

.create-post_header {
  width: 100%;
  display: flex;
  flex-flow: row nowrap;
  align-items: center;
  justify-content: center;

  box-sizing: border-box;
  padding: 16px;
}

.header_info {
  width: 100%;
  height: 100%;
  display: flex;
  flex-flow: column nowrap;
  align-items: flex-start;
  justify-content: center;
  padding-left: 16px;
}
```



## CreatePostFieldText

### Description

#### Issue(s)

- Add

#### Parameters

Name	Type	Description
Type name	Type type	Type description

#### Source code

**\*\*\*Double click to view full page.**

```
import './CreatePostFieldText.css';
```

```
const CreatePostFieldText = ({ content, handlePostChange }) => {  
  return (  
    <textarea  
      className='container__post-content'  
      rows={10}  
      placeholder={'What's on your mind?'}  
      name='content'  
      value={content}  
      onChange={handlePostChange}  
    />  
  );  
};  
  
export default CreatePostFieldText;
```

## CSS

**\*\*\*Double click to view full page.**

```
.container__post-content {
  outline: 0;
  border: 0;
  margin: 0;
  padding: 0;

  width: 100%;
  height: fit-content;
  border-radius: 0 !important;
  background-color: transparent;

  font-size: 16px;
  font-family: 'Varela Round', sans-serif !important;
}

.container__post-content > textarea {
  font-family: 'Varela Round', sans-serif !important;
}

::-webkit-input-placeholder {
  font-family: 'Varela Round', sans-serif !important;
}

:-moz-placeholder {
  font-family: 'Varela Round', sans-serif !important;
}

::-moz-placeholder {
  font-family: 'Varela Round', sans-serif !important;
}

:-ms-input-placeholder {
  font-family: 'Varela Round', sans-serif !important;
}
```

## CreatePostSelectPrivacy

### Description

#### Issue(s)

- Add

#### Parameters

Name	Type	Description
Type name	Type type	Type description

#### Source code

**\*\*\*Double click to view full page.**

```
import './CreatePostSelectPrivacy.css';
```

```
const DEFAULT_PRIVACY_OPTIONS = ['Friends', 'Only Me', 'Public'];
```

```
const CreatePostSelectPrivacy = ({ privacy, handlePostChange, options = DEFAULT_PRIVACY_OPTIONS }) => {  
  const privacyOptions = options.map((option) => <option value={option}>{option}</option>);
```

```
  return (  
    <select className="container__post-privacy" name="privacy" id="privacy"  
value={privacy} onChange={handlePostChange}>  
      {privacyOptions}  
    </select>  
  );  
};
```

```
export default CreatePostSelectPrivacy;
```

CSS

**\*\*\*Double click to view full page.**

```
.container__post-privacy {  
  border: 0;  
  outline: 0;  
  background-color: transparent;  
  width: 112px;  
  height: 32px;  
  font-size: 16px;  
  letter-spacing: 2px;  
}  
  
.container__post-privacy > option {  
  font-size: 16px;  
  height: 32px;  
  padding: 8px;  
  margin: 16px;  
  box-sizing: border-box;  
}
```

## CreatePostUserAvatar

### Description

#### Issue(s)

- Add

#### Parameters

Name	Type	Description
Type name	Type type	Type description

#### Source code

\*\*\*Double click to view full page.

```
import './CreatePostUserAvatar.css';
```

```
const CreatePostUserAvatar = ({ imagePath }) => {  
  return <img className="container_post-avatar" src={imagePath} alt="imagePath"  
/>;  
};
```

```
export default CreatePostUserAvatar;
```

#### CSS

\*\*\*Double click to view full page.

```
.container_post-avatar {  
  width: 48px;  
  height: 48px;  
  border-radius: 50%;  
  object-fit: cover;  
  overflow: hidden;  
}
```

## CreatePostUserName

### Description

#### Issue(s)

- Add

#### Parameters

Name	Type	Description
Type name	Type type	Type description

#### Source code

**\*\*\*Double click to view full page.**

```
import './CreatePostUserName.css';
```

```
const CreatePostUserName = ({ user }) => {  
  const fullName = `${user.firstName} ${user.lastName}`;  
  return <p className="container_post-username">{fullName}</p>;  
};
```

```
export default CreatePostUserName;
```

#### CSS

**\*\*\*Double click to view full page.**

```
.container_post-username {  
  font-size: 16px;  
  font-weight: bold;  
  letter-spacing: 2px;  
  font-family: 'Varela Round', sans-serif;  
}
```

## CreatePostBodyOptions

### Description

#### Issue(s)

- Add

#### Parameters

Name	Type	Description
Type name	Type type	Type description

#### Source code

**\*\*\*Double click to view full page.**

```
import './CreatePostBodyOptions.css';
```

```
const CreatePostBodyOptions = ({ children }) => {  
  return <div className="container__body-options">{children}</div>;  
};
```

```
export default CreatePostBodyOptions;
```

#### CSS

**\*\*\*Double click to view full page.**

```
.container__body-options {  
  width: 98%;  
  height: 48px;  
  border-radius: 8px;  
  border: 2px solid #b89f9f;  
  
  display: flex;  
  justify-content: flex-end;  
  align-items: center;  
  flex-flow: row nowrap;  
  box-sizing: border-box;  
  padding: 8px 16px;  
}
```

## CreatePostImages

Description

Issue(s)

- Add

Parameters

Name	Type	Description
Type name	Type type	Type description



Source code

**\*\*\*Double click to view full page.**

```
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faCircleNotch } from '@fortawesome/free-solid-svg-icons';

import './CreatePostImages.css';

const CreatePostImages = ({ fileSelected = [] }) => {
  const showFilesSelected = fileSelected.files.map((image, index) => {
    if (image.isReading)
      return (
        <span key={index} className="container__images_loading">
          <span>
            <FontAwesomeIcon icon={faCircleNotch} />
          </span>
        </span>
      );
    return (
      <div className="container__images_img">
        <img
          key={index}
          src={image.preview}
          alt={`preview${index}`}
          onClick={() => {
            fileSelected.fileDeleteByIndex(index);
          }}
        />
      </div>
    );
  });

  const isShowRemoveButton = fileSelected.files.length > 0 && (
    <button className="post-container__images_button-close" onClick={fileSelected.fileDeleteAll}>
      Remove all
    </button>
  );

  return (
    <div className="post-container__images">
      {isShowRemoveButton}
      {showFilesSelected}
    </div>
  );
};

export default CreatePostImages;
```

CSS

\*\*\*Double click to view full page.

```
.post-container__images,
.container__images_img,
.container__images_loading {
  display: flex;
  flex-flow: row nowrap;
  align-items: center;
  justify-content: center;
}

.post-container__images {
  width: 100%;
  height: fit-content;
  overflow-x: auto;
  align-items: flex-start;
  justify-content: flex-start;
  position: relative;
  padding: 8px 8px;
  box-sizing: border-box;
}

.post-container__images_button-close {
  position: sticky;
  top: 0;
  left: 0;
  outline: 0;
  border: 0;
  border-radius: 8px;
  min-width: 120px;
  min-height: 32px;
  font-size: 16px;
  text-align: center;

  color: #fff;
  background-color: #00000034;
}

.container__images_loading {
  width: 160px;
  height: 160px;
}

.container__images_loading > span {
  font-size: 16px;
  animation: anim-loading 1s infinite ease-in-out;
}

.container__images_img {
  min-width: 140px;
  height: 90px;
  margin: 0 4px;
}
```

## FileSelect

### Description

### Issue(s)

- Add

### Parameters

Name	Type	Description
Type name	Type type	Type description

### Source code

\*\*\*Double click to view full page.

```
import { useRef } from 'react';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faFileImage } from '@fortawesome/free-solid-svg-icons';

import './FileSelect.css';

const FileSelect = ({
  buttonSelectText = <FontAwesomeIcon icon={faFileImage} />,
  multipleFiles = true,
  acceptFiles = 'image/*',
  onfileSelect = (event) => {},
}) => {
  const fileInputRef = useRef();

  const onFileClick = (event) => {
    event.preventDefault();
    fileInputRef.current.click();
  };

  return (
    <div className="container__file-select">
      <button onClick={onFileClick}>{buttonSelectText}</button>

      <input
        ref={fileInputRef}
        type="file"
        multiple={multipleFiles}
        style={{ display: 'none' }}
        accept={acceptFiles}
        onChange={onfileSelect}
      ></input>
    </div>
  );
};

export default FileSelect;
```

CSS

**\*\*\*Double click to view full page.**

```
.container__file-select {  
}  
  
.container__file-select > button {  
  outline: 0;  
  border: 0;  
  background-color: transparent;  
}
```

useFileSelect

Description

Issue(s)

- Add

Parameters

Name	Type	Description
Type name	Type type	Type description

Source code

\*\*\*Double click to view full page.

```
import { useMemo, useState } from 'react';

const fileSchema = { file: '', preview: '', isReading: false };
const useFileSelect = () => {
  const [files, setFiles] = useState([]);
  const filesCount = useMemo(() => {
    return files.length;
  }, [files]);

  const preparePreviewFile = (index, fileToUpdate = fileSchema) => {
    const reader = new FileReader();

    reader.onload = () => {
      console.log('reader onload');

      fileToUpdate.preview = reader.result;
      fileToUpdate.isReading = false;
      fileUpdateByIndex(index, fileToUpdate);
    };

    reader.readAsDataURL(fileToUpdate.file);
  };

  const fileDeleteByIndex = (index) => {
    setFiles((currentFiles) => currentFiles.filter((file, fileIndex) => fileIndex
    !== index));
  };

  const fileDeleteAll = () => {
    setFiles([]);
  };

  const fileUpdateByIndex = (index, fileUpdate = fileSchema) => {
    setFiles((currentFiles) => {
      let newFiles = [...currentFiles];
      newFiles[index] = fileUpdate;
      return newFiles;
    });
  };

  const filesAdd = (files = new FileList()) => {
    let fileIndex;

    for (fileIndex = 0; fileIndex < files.length; fileIndex++) {
      const currentFile = files[fileIndex];
      if (currentFile && currentFile.type.substring(0, 5) === 'image') {
        const newFile = { file: currentFile, preview: '', isReading: true };
        preparePreviewFile(fileIndex, newFile);
        setFiles((currentFiles) => [...currentFiles, newFile]);
      }
    }
  }
}
```

useCreatePost

Description

*ComponentName*

Issue(s)

Parameters

Name	Type	Description
Type name	Type type	Type description

Source code

\*\*\*Double click to view full page.

```
import { useEffect, useState, useContext } from 'react';
import { useHistory } from 'react-router-dom';

import UserService from '../services/UserService';
import UserContext from '../contexts/UserContext';
import SwapService from '../services/SwapService';
import AuthService from '../services/auth.services';

const postResponseScheme = { ok: false, loading: false, message: '', result: null
};
const useCreatePost = () => {
  let history = useHistory();

  const currentUser = useContext(UserContext).user;
  const [user, setUser] = useState([]);
  const [post, setPost] = useState({ privacy: 'Public', content: '' });
  const [postResponse, setPostResponse] = useState({ ...postResponseScheme });

  const postUpdate = ({ name, value }) => {
    const changedProp = { [name]: value };
    setPost((previousValue) => {
      return { ...previousValue, ...changedProp };
    });
  };

  const postReset = () => {
    setPost({ privacy: 'Public', content: '' });
  };

  const postUpload = async (files = [], isSwap = true) => {
    setPostResponse({ ok: false, loading: true, message: 'Posting...', result:
null });

    const hasUpdatedContent = post.content.length > 0;
    const hasImages = files.length > 0;

    if (isSwap) files = files[0];
    if (!hasImages) return setPostResponse({ ok: false, loading: false, message:
'No images found.', result: null });
    if (!hasImages && !hasUpdatedContent)
      return setPostResponse({ ok: false, loading: false, message: 'No images and
content found.', result: null });

    const formData = new FormData();
    formData.append('content', post.content);
    formData.append('files', files);
    formData.append('privacy', post.privacy);

    try {
      const swapResponse = await SwapService.createSwap(user.id, formData);
```