For this, we will be using a library called React Native Fetch Blob. This library does not work with Expo so we will not be using Expo.

**-** We start a new React project and initialize RN Fetch Blob

npm i rn-fetch-blob

**-** We also import it

import RNFetchBlob from 'rn-fetch-blob';

-Next, add external storage permission in android/app/src/AndroidManifest.xml

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

-We create a *useState(img\_name, setImgName)* and a handler for it *handleImgName*. This handler will use the name passed to it to find our image from gallery and display.

-Create another useState(show, setShow) which is initially set to false. Our downloaded image is displayed whenever this state is true

-Next, we get permission from the user to save the image. If permission is granted, we call the *downloadImage()* async function which will return the generated image name. When we get this name, we wait for 5 seconds to enable the image to save properly, then pass the name to *handleImgName,* finally *setShow* is set to true so downloaded image can be displayed.

The *downloadImage():*

**-**We get the filesystem(fs) and config from rn-fetch-blob.

const {config, fs} = RNFetchBlob

**-**With the fs, we create a new directory inside the pictures directory folder(PictureDir). The PictureDir is part of a collection of commonly used dirs that come with rn-fetch-blob.

const newDir = dirs.PictureDir+ '/reactsavedata';

fs

.mkdir(newDir)

.catch(err => {

console.log(err);

});

A random name for the image is then generated using the current time and date, we also get the image extension from the URL, and these are used to generate the image path

const randd = Math.floor(date.getTime()+date.getSeconds()/2)

const generated\_img\_path = PictureDir + '/image\_' + randd + ext

**-**With the config, we configure the download options: which consist of setting the image path, download notifications etc.

Finally, we return the generated image name so the *handleImgName* can use it.