Practical Test

Your app should have the following screens – Albums, Photos and Wishlist. Albums and Wishlist screens should be in a tabbar.

Important Notes:
- **Implement redux-thunk for state management if you know it otherwise use redux-saga.**
- **For icons and other assets, you can use relative images from the internet and can use react-native vector icons.**
- **Proper validations, responsive UI across various devices, accuracy in the design etc needs to be taken care of.**
- **Please read the paper carefully and ask for any queries.**
- **Share the completed work in a zip with us once you are done.**

1. Albums
- Show the list of albums with infinite scrolling and pull to refresh functionality.
- Get Albums list by call to api using below url - 'https://jsonplaceholder.typicode.com/albums'.

Important: This list needs to be fetched every time, user opens the app.. Utilise helper functions to call this api.

- Append the same album list to itself to get data for infinite scrolling. On press of album it will navigate to the next screen which is **Photos.**
- Show Id and title of album as per design

Albums

1

consectetur vel rerum qui aperiam modi eos asr ewq… →

2

consectetur vel rerum qui aperiam modi eos asr ewq… →

3

consectetur vel rerum qui aperiam modi eos asr ewq… →

4

consectetur vel rerum qui aperiam modi eos asr ewq… →

5

consectetur vel rerum qui aperiam modi eos asr ewq… →

2. Photos

- Design the screen as mentioned in the following image.
- Call this API for getting the list of images: 'https://jsonplaceholder.typicode.com/album/2/photos' (pass the album ID instead of 2)
- All the data needs to store in redux
- Show Image title and url from the response
- Append the same photos list to itself to get data for infinite scrolling as well as implement the pull to refresh which will fetch the data from scratch.
- Implement the local search on the fetched result.
- On press of wishlist icon it will add/remove from wishlist. By default none of the item will be in wishlist
- Wishlisted items needs to be shown in wishlist screen

← Photos

Search by title

sunt quo laborum commodi te...

sunt quo laborum commodi te...

3. Wishlist

- This is second tab screen: Need show all the wishlisted items
- User can remove from wishlist
- This list needs to be persisted so until and unless user delete the application then data will be removed
- List data needs to be fetched from redux store.