


Zepz Assessment

Background

- The project was re-initialized due to issues with installing all required plugins and software on an M1 Mac.
- Typescript was chosen to help with typechecking and to reduce runtime errors.
- The project is structured so that re-usable items can be easily accessed where needed. Each component is in its own folder, and screens are in the screens folder.
- The Utils folder contains global constants, colors, and any utilities used throughout the app.
- The models folder houses the app state, which uses zustand under the hood. The reason for using zustand was because it does not tie the entire app or a section of the app to a context, which can sometimes cause the zombie child effect. It also comes with a persist plugin that allows for saving state to local storage, allowing for saving the blocked and following list even after reloading the app.

Setting up

- The uses react native to startup and run the app follow the below
 - `cd` into the app directory
 - run `yarn install`
 - `cd ios && pod install` (to install cocoa-pods)
 - `cd ..` (to go back to the directory)
 - `yarn start` to (start the metro server)
 - open another terminal and run:
 - `npx react-native run-ios` for  iPhone's
 - `npx react-native run-android` for android phones
 - this will open a simulator

Challenges

- Running the provided project on my M1 mac.
- Enough time to refine routing and add extra bits.

Improvements

- Paginating items
- Having an infinite scroll
- different route for favourite items

Testing

- The app utilizes Zustand for state management, along with a persist middleware from Zustand to cache favorites and maintain them when the user reloads the app.
- I used the react-testing library to write unit tests.
- Although I would have liked to have at least one end-to-end test, the current tests are done using @testing-library.
 - The current tests are located in `src/screens/home/components/user-list-item/__tests__`.
 - Currently, the tests cover:
 1. Expanding a list item
 2. Following a user
 3. Blocking a user