

Code Review written by Yevgeny Ivanov 324780246
Chosen file: SignUp.tsx

Code Readability:

- Little to no documentation
- Styling in few instances present in the Touchables and not in the StyleSheet below
- Function names are clear
- Few unnecessary imports
- Code structure is consistent: imports, components, functions, on-screen display, styling
- The component is well-structured and follows React's functional component pattern with hooks.
- Proper use of the React Native components.\
- authentication logic handled by an outside class. Keeps the code clean, read-able and more maintainable.
- Multiple authentication options : Facebook, Google, E-mail

Improvements:

- Remove unused imports
- move 'TouchableOpacity' styling to the style sheet below
- add documentation in key parts of the code, ie: brief async function description or a brief description of a 'View'
- Since this is React Native, `signInWithPopup` won't work (it only works in web applications). We may need to use `react-native-google-signin`.

Bugs and Potential Future Issues:

- Google Auth function leads to "Main" while Facebook Auth leads to "Login". Handling both cases as an external sign in that leads to the same page would be more optimal
- Sign Up doesn't directly lead to the main screen, rather goes to the "Login" screen. In real applications/websites after signing up the user is shown the main screen.
- Hardcoded values for the pages names:

```
if (result.success) {  
  navigation.navigate("Main");  
  Alert.alert('Success', result.messa
```

This can lead to problems down the line if the names of the pages will be changed.
A good solution may be a definition of a const, like this:

```
export const ROUTES = {  
  MAIN: "Main",  
  LOGIN: "Login",  
};
```

and called like this:

```
navigation.navigate(ROUTES.LOGIN);
```

- Missing input validation on the front-end before calling `handleSignUp` function.