Code Review written by Yoni Baruch 211465786

Chosen file: CreateRequestPage.tsx

Code Readability:

- The code is barely documented.
- There are some mistakes in names of states such as 'phonenumberm' instead of 'phonenumber' and 'DestinationLoaction' instead of 'location'. These are just bugs waiting to happen.
- Function names are clear are great, although onCurrentPlaceChanged could be shortened.
- When catching errors, Alerts (pop-ups) don't provide insightful messages in order for the user to correct himself.

Improving Structure:

- Overall structure of the code 'import, components, functions and styles..' is a common structure for react native projects.
- Styling names are up to standard.
- There is a loading screen which is nice.
- The 'current location' and 'destination location' UI code are extremely similar can be moved into a separate component called 'location' since it is reused in the UI.

Bug Detection and Efficiency:

- The Google API key is exposed in the code, this is a major security issue and should be stored somewhere else.
- handleSubmit doesn't check for nulls or any of the fields. We can send out an empty request to the writer which is clearly not intended.
- in 'fetchCurrentLocation', location isn't validated before updating the state to it. Can cause error or an empty UI space.
- No 'keyExtractor' in ScrollView - this could help react native to optimize when rendering the list.

Coding Standards:

- 'fetchCurrentLocation' is only called when the component mounts. There should be button to refresh the current location - there could be issues with the location API that would require a user to want to re-fetch his location.
- There is no validation in any field.
- As we said, it's a common coding convention to place the API key in an env. variable.
- There should be a function to clear all fields.
- in 'handleSubmit' current states should be a entered as arguments instead of the function using them directly. Its a bad standard since the states can be out of sync while the function is executing. When they are fed into the argument, it acts as a "snapshot" of the states and is much more safer.

3 Key comments:

- **Validation:** Important fields like title, phone number, and destination need required checks with proper formats. Real-time feedback helps users fix mistakes quickly, making the app more reliable.
- **API Key:** Hardcoding the API key is unsafe as it can be stolen and misused. It should be stored securely with access restrictions to prevent unauthorized use.
- **Alerts:** Current alerts are vague. Clear, specific messages like “Invalid phone number” help users understand and fix issues faster.

Example of Improvement:

```
const handleSubmit = async () => {
  if (validateFields()) {
    Alert.alert("Validation Error", "Please fill all the required fields.");
    return;
  }
  try {
    await writer.addRequest(
      title,
      currentCoordinates ? `${currentCoordinates.latitude},${currentCoordinates.longitude}` : '',
      currentAddress,
      DestinationLocation,
      additionalNotes,
      phoneNumber
    );
    Alert.alert("Success", "Request added successfully");
    console.log("Request added successfully");
    clearAllFields();
    Alert.alert("Success", "Request added successfully");
  } catch (error) {
    Alert.alert("Error", "Failed to add request");
  }
};
```