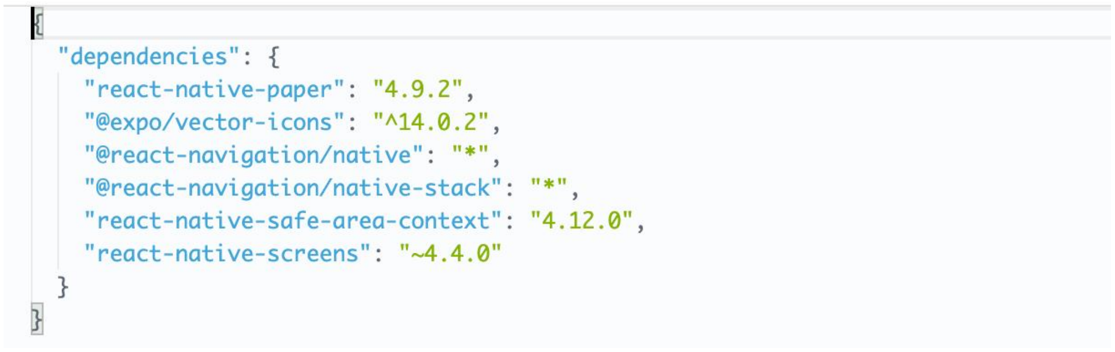
ReadMe

1. how to configure and run the project:

Github link: https://github.com/wsfs666/MP_exercise2

Snack link: https://snack.expo.dev/@sfu13/exercise

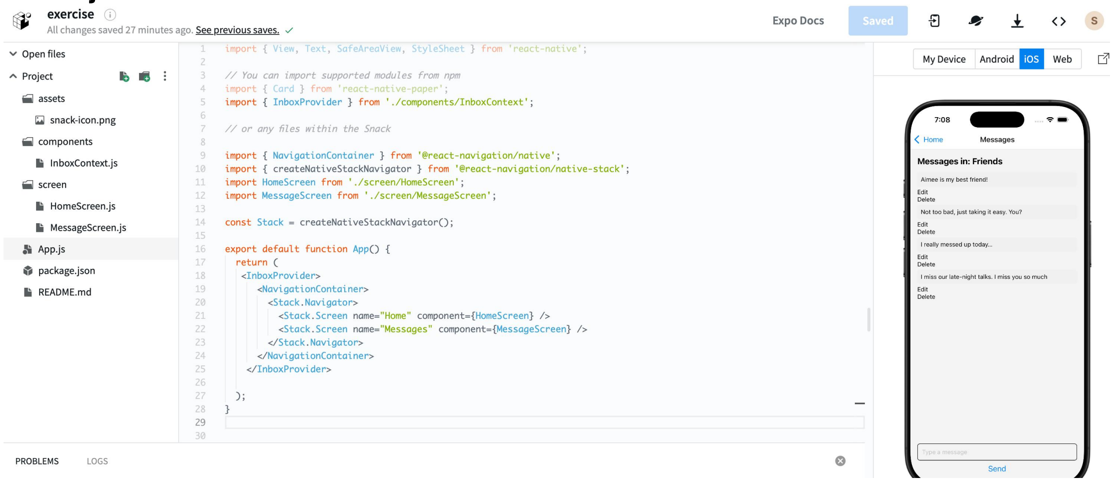
Configure package.json



```
{
  "dependencies": {
    "react-native-paper": "4.9.2",
    "@expo/vector-icons": "^14.0.2",
    "@react-navigation/native": "*",
    "@react-navigation/native-stack": "*",
    "react-native-safe-area-context": "4.12.0",
    "react-native-screens": "~4.4.0"
  }
}
```

I run the project in the online expo snack playground. So I just click on “Launch snack” to run ths project in IOS simulator.

2. Project structure:



```
import { View, Text, SafeAreaView, StyleSheet } from 'react-native';

// You can import supported modules from npm
import { Card } from 'react-native-paper';
import { InboxProvider } from './components/InboxContext';

// or any files within the Snack

import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import HomeScreen from './screen/HomeScreen';
import MessageScreen from './screen/MessageScreen';

const Stack = createNativeStackNavigator();

export default function App() {
  return (
   <InboxProvider>
      <NavigationContainer>
        <Stack.Navigator>
          <Stack.Screen name="Home" component={HomeScreen} />
          <Stack.Screen name="Messages" component={MessageScreen} />
        </Stack.Navigator>
      </NavigationContainer>
    </InboxProvider>

  );
}
```

7:08
Home Messages

Messages in: Friends

Aimee is my best friend!
Edit
Delete

Not too bad, just taking it easy. You?
Edit
Delete

I really messed up today...
Edit
Delete

I miss our late-night talks. I miss you so much
Edit
Delete

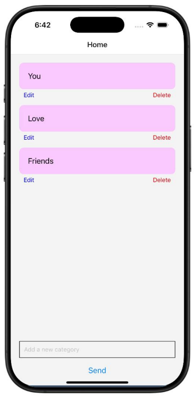
Type a message
Send

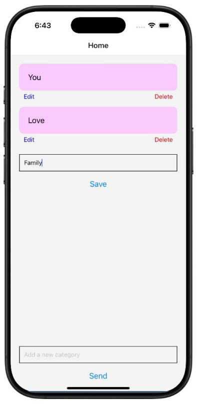
In this project, I designed a two-page navigation structure. The first page is the Home screen, which displays different message categories. The second page is the Messages screen, showing all messages under a selected category. This app supports full CRUD functionality for both message categories (inboxes) and individual messages.

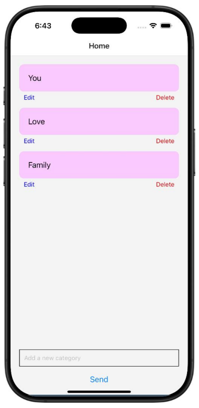
To manage shared state, I created an InboxContext.js file. The InboxContext is implemented using the React Context API and useState. It acts as a global state manager, allowing different components to access and update inbox and message data without manually passing props. All CRUD operations for both inboxes and messages are handled through context methods, ensuring that changes are reflected across the entire app in real time.

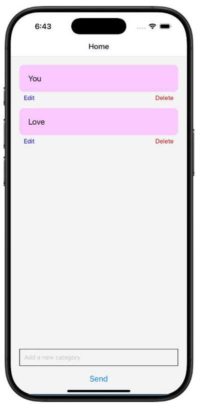
The HomeScreen displays a list of message categories (inboxes), allowing users to add, edit, delete, or tap a category to view its messages. The MessagesScreen shows all messages under the selected inbox, where users can create, edit, and delete individual messages. Both screens use shared state from InboxContext to keep inbox and message data synchronized across the app.

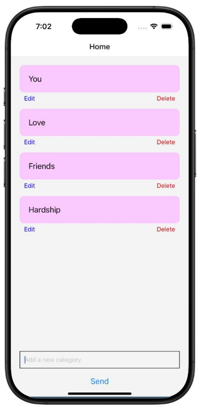
3. **Screenshots:**



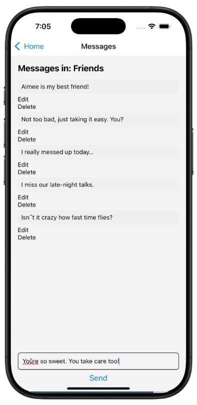








The above 5 screenshots show the feature of deleting a category, updating a category and adding a category in HomeScreen.js. The HomeScreen displays all inbox categories using a FlatList. Users can add a new category through a TextInput and a "Send" button. Each category can be edited inline by toggling to an input field with a "Save" button, or deleted using a "Delete" button. When a category is tapped, it navigates to the Messages screen with the corresponding inbox ID and title. All operations (add, edit, delete) update the global inbox state via methods provided by InboxContext.



Messages in: Friends

Aimee is my best friend!

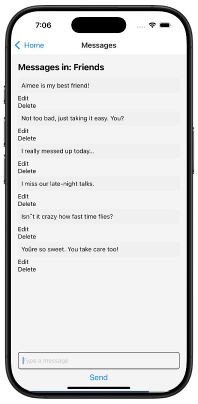
Not too bad, just taking it easy. You?

I really messed up today...

I miss our late-night talks.

Isn^t it crazy how fast time flies?

Yoûre so sweet. You take care too!



Messages in: Friends

Aimee is my best friend!

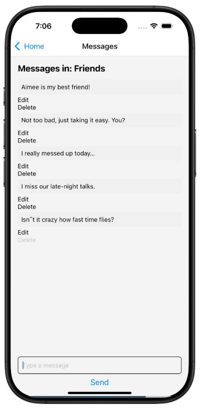
Not too bad, just taking it easy. You?

I really messed up today...

I miss our late-night talks.

Isn^t it crazy how fast time flies?

Yoûre so sweet. You take care too!



Messages in: Friends

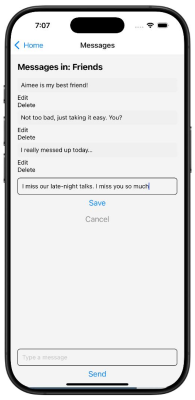
Aimee is my best friend!

Not too bad, just taking it easy. You?

I really messed up today...

I miss our late-night talks.

Isn^t it crazy how fast time flies?



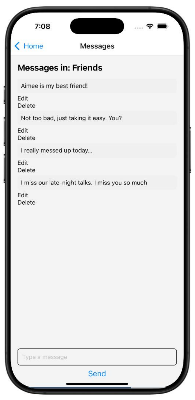
Messages in: Friends

Aimee is my best friend!

Not too bad, just taking it easy. You?

I really messed up today...

I miss our late-night talks. I miss you so much



Messages in: Friends

Aimee is my best friend!

Not too bad, just taking it easy. You?

I really messed up today...

I miss our late-night talks. I miss you so much

The above 5 screenshots show the feature of deleting a message, updating a message and adding a message in MessageScreen.js. The MessagesScreen shows all messages under the selected inbox, also using a FlatList. Users can add a new message at the bottom using an input field and "Send" button. Each message can be edited inline or removed. Editing toggles a TextInput with "Save" and "Cancel" buttons. All message updates are scoped to the current inbox and handled by methods in InboxContext, keeping data consistent across screens.