For this task, you’ll store your chat app’s messages offline using React Native’s AsyncStorage. In your [Achievement 5 project brief](https://images.careerfoundry.com/public/courses/fullstack-immersion/full-stack-project-briefs/A5-Project-Brief-Mar2023.pdf), you can find the following user story that relates directly to this task:

“As a user, I want to be able to read my messages offline so I can reread conversations at any time.”

**Directions**

**Step 1.** In "App.js" implement a real-time network connectivity detection system.

* Use useNetInfo().
* Disable Firestore when there’s no connection and enable it otherwise (involves disableNetwork() and enableNetwork()).
* Pass a prop isConnected (represents the connection status) to the Chat component.

**Step 2.** In "Chat.js", utilize the isConnected prop to decide what source to fetch data from.

* If there’s a connection, fetch messages from the Firestore database and cache messages whenever possible in an onSnapshot callback function (involves AsyncStorage.setItem()).
* Otherwise, load cached messages from the local storage (involves AsyncStorage.getItem()).

**Step 3.** When offline, prevent Gifted Chat from rendering the InputToolbar so users can’t compose new messages. InputToolbar is the UI part of Gifted Chat that contains an input field and the "Send" button.

Note!  
To change Gifted Chat’s InputToolbar, override the prop renderInputToolbar={...} of the <GiftedChat …/> component the same way you changed the renderBubble prop in task 5.2:

* Import InputToolbar from the "react-native-gifted-chat" package.
* Create a new function that returns <InputToolbar {...props}/> if there’s a connection, otherwise, it returns a null:

const renderInputToolbar = (props) => {

if (isConnected) return <InputToolbar {...props} />;

else return null;

}

**Step 4.** Test your app on- and offline using expo start --offline. Refer to the “Testing the App Offline” section of the Exercise to review the steps needed to test your app’s offline features.

**Step 5.** Submit the GitHub repo of your app here for your tutor to review.

**Step 6.** Upload a recording of your app while using it. Make sure to showcase how your app’s chat screen behaves while the emulator is in offline mode.

* Watch this [Google tutorial](https://developer.android.com/studio/debug/am-video#record-video-with-the-emulator) if you need a refresher on recording your emulator’s screen).
* The app should render messages retrieved from AsyncStorage when in offline mode.
* To stop users from composing new messages, the InputToolbar shouldn’t render while the app’s in offline mode.