ReadMe File

Exercise Title

Chating_with_Loved1s

Course Information

- **Course Code:** CS5450 – Mobile Programming
- **Instructor:** Dr. Sabah Mohammed
- **Institution:** Lakehead University

Introduction

Chat_with_Loved1s is a modern and visually appealing mobile chat application developed using **React Native** and **Expo**. Inspired by popular messaging platforms like WhatsApp, this app provides users with a smooth, feature-rich chatting experience in a soft, pastel-themed UI.

The app is designed to simulate real-time conversations with preloaded contacts, offering essential functionalities such as message sending, editing, deleting, pinning, and read indicators. With a focus on clean design and responsive layout, the application ensures usability across both iOS and Android platforms.

Whether you're testing UI/UX design or exploring mobile app development using React Native, *Chat_with_Loved1s* is a practical and aesthetic showcase of chat application fundamentals.

Feature

The application provides the following features:

1. **Preloaded Contacts**
   Includes 5 default contacts (Jaimin, Shlok, Myntra, Zeel, and Laksh), each with profile photos.
2. **Add New Contact**
   Users can dynamically add new contacts with a single tap.
3. **One-on-One Messaging**
   - Send messages instantly
   - Auto-reply feature mimics real-time conversation
4. **Message Actions**
   - **Edit:** Update sent messages
   - **Delete:** Remove messages
   - **Pin/Unpin:** Highlight important messages
   - **Seen Tick:** Double-tick indicators for message status
5. **Message Metadata**

- Time of message
- Edited tag
- Pin icon
- Read receipts

6. **Smooth UI/UX**
   - Light pink theme with elegant layout
   - Responsive design for Android and iOS
   - Keyboard avoids input box to maintain usability
7. **Custom Modal for Actions**
   Long-pressing a message opens a clean modal for message options.

App design & structure

Overall Design Aesthetic
The app follows a light and friendly theme using a **soft pink background** and **white message bubbles** with **black text** for clear readability. The UI is minimal, smooth, and closely resembles modern chat apps like WhatsApp.

2. Main Screens

- **Contact List Screen:**
  Displays preloaded contacts in styled pink cards with avatars and names. Includes a header titled `Chat_with_Loved1s` and a button to add new contacts.
- **Chat Screen:**
  A dedicated chat window for each contact:
  - Pink header bar with avatar and back button
  - Message area with sent and received messages
  - Rounded input field fixed at the bottom with a send icon
  - Responsive to virtual keyboard

3. Component Breakdown

- `App.js`: Main logic and layout handling
- `ContactList`: Scrollable view of all available contacts
- `MessageItem`: Reusable component to render individual messages with styles and metadata
- `Modal`: Custom modal for edit, delete, and pin actions
- `KeyboardAvoidingView` & `SafeAreaView`: Ensure content doesn't overlap with keyboard or device UI

4. Styling

- Centralized in `StyleSheet.create()`

- Uses pastel shades for backgrounds and accent buttons
- Rounded corners, consistent padding, and visual hierarchy for better UX

5. Responsiveness

- Works across Android and iOS using Expo
- Handles keyboard appearance and safe zones
- Fully scrollable and adaptable layout

Technologies used

- **React Native**
  Core framework used to build the cross-platform mobile application.
- **Expo (Snack.expo)**
  Simplifies development and testing without native build tools. Enables quick live previews on Android and iOS.
- **JavaScript (ES6+)**
  Main programming language used to create the app's logic and components.
- **React Hooks (`useState`, `useEffect`)**
  For managing state, lifecycle events, and UI reactivity.
- **React Native Components**
- `View`, `Text`, `TouchableOpacity`, `TextInput`, `FlatList`, etc., for building UI.
- `KeyboardAvoidingView` and `SafeAreaView` for mobile-friendly layouts.
- **Expo LinearGradient**
  Adds smooth background gradients for visual polish.
- **@expo/vector-icons (Ionicons)**
  Used for navigation icons, send button, and action icons like edit, delete, and pin.
- **Custom Modal System**
  Built-in React Native `Modal` component for message action handling.

Setup & Installation

You can run this app directly using **Expo Snack**—no installation required!

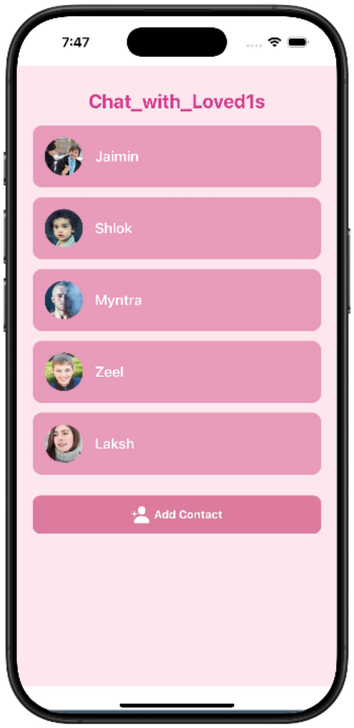
- **Run Online (Recommended)**

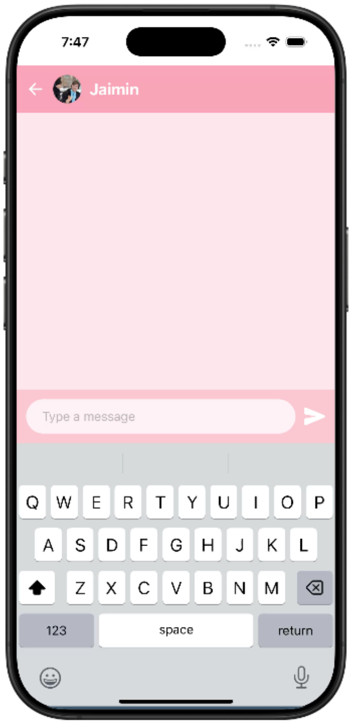
1. Go to: https://snack.expo.dev
2. Write the full app code into the editor
3. Click **Run on iOS** or **Run on Android**
4. Or scan the QR code using the **Expo Go** app on your mobile device

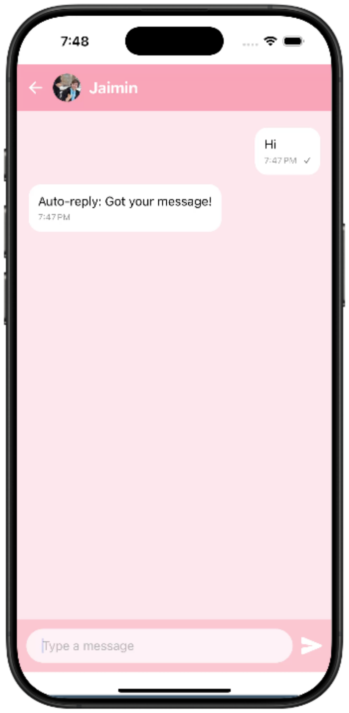
Usage

1. **Launch the App**
   Open the app in Expo (Snack.expo or Expo Go on your device).
2. **View Contacts**
   You'll see 5 preloaded contacts with their profile pictures and names.
3. **Start a Chat**
   - Tap on any contact to open a dedicated chat screen
   - The contact's photo and name appear at the top
4. **Send Messages**
   - Type your message in the input box
   - Press the send icon to deliver it
   - A **fake auto-reply** will be received after 1 second
5. **Use Message Options**
   - **Long press** on your sent message to open the action modal
   - Choose to **Edit**, **Delete**, or **Pin/Unpin** the message
   - **Pinned messages** stay at the top of the list
   - **Edited** messages show an "edited" tag
   - **Seen tick** appears as ✓✓ when the bot has seen your message
6. **Add New Contacts**
   - Scroll down the contact list
   - Tap "Add Contact" to instantly add a new contact with a generated avatar and name

Output

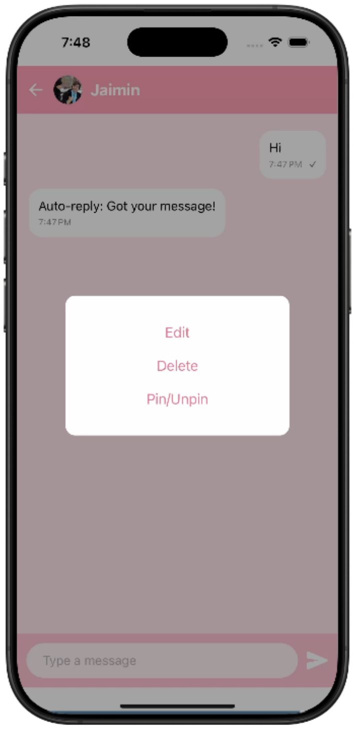






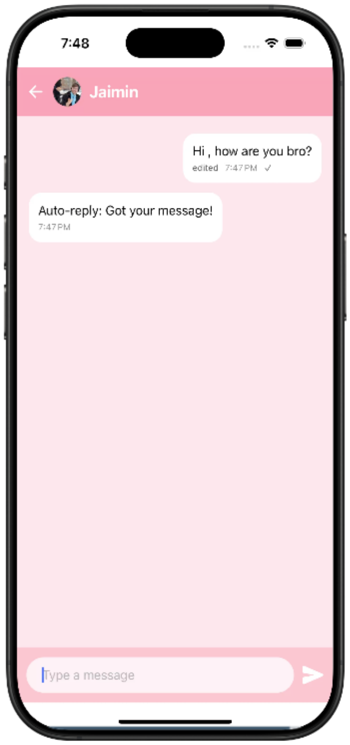
Hi
7:47 PM ✓

Auto-reply: Got your message!
7:47 PM



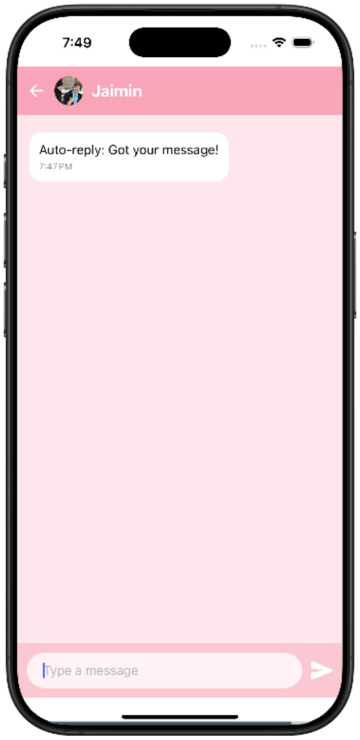
Hi
7:47 PM ✓

Auto-reply: Got your message!
7:47 PM



Hi , how are you bro?
edited 7:47 PM ✓

Auto-reply: Got your message!
7:47 PM



Auto-reply: Got your message!
7:47 PM