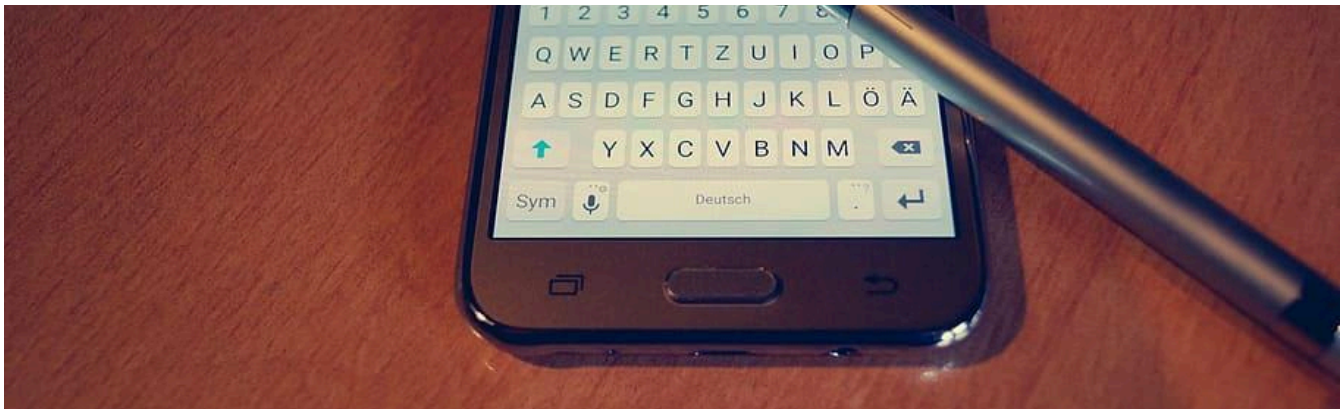
Chat App Case Study



ExecutiveSummary

A mobile chat app for Android and iOS that lets users message in real time, share photos, and even send their location.

Introduction

This project was designed to explore mobile development using React Native. The goal was to build a real-time chat app that works smoothly on both iOS and Android. The app lets users pick a name, choose a chat background, and send messages with text, photos, or their current location. It also works offline and stores messages locally.

The Problem

Many chat apps are either too complex or require accounts and sign-ins that create friction for new users. I wanted to build a lightweight mobile app that lets people start chatting quickly, share rich media, and stay connected even when they lose service.

My Role

I built the entire application using React Native and Expo. I connected it to Firebase to manage real-time data, media storage, and user sessions. I focused on both the interface and the behind-the-scenes features that support fast, smooth messaging.

Timeline

This project took around 3 weeks to complete, from setup through testing on physical devices.

Tools and Skills Used

Frameworks and Tools:

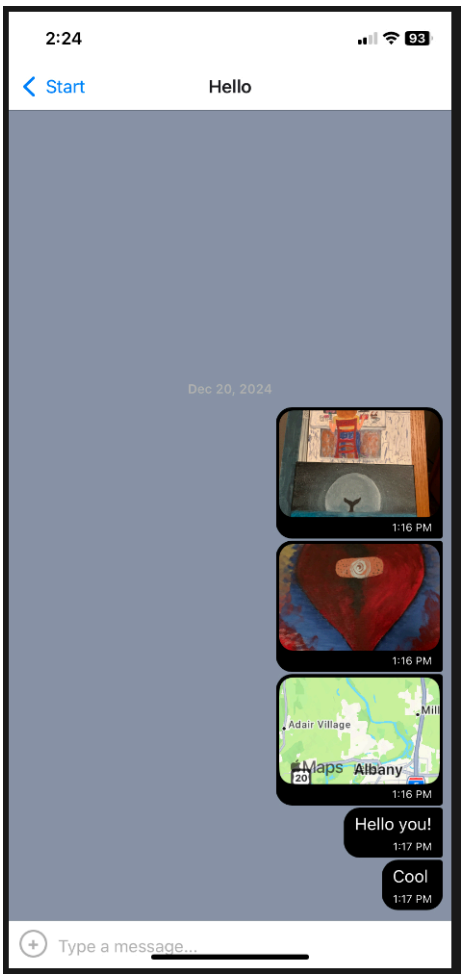
- React Native - for building the app on both Android and iOS
- Expo - to simplify mobile development and testing
- Firebase - for real-time database, media storage, and user authentication

Libraries and Features:

- react-native-gifted-chat for the chat interface
- expo-location for location sharing
- Expo-image-picker for selecting or taking photos
- @react-native-async-storage/async-storage for offline storage of messages

Skills I practiced:

Mobile development, real-time data syncing, working with device permissions, and creating offline-first apps.



Key findings

Key Finding 1:

Users appreciated being able to message without needing to sign up or log in. The anonymous login removed friction.

Key Finding 2:

Being able to share pictures from the camera or photo library added a personal touch to conversations.

Key Finding 3:

Location sharing made the app more interactive and showed how real-time features can be more than just text-based.

Solutions

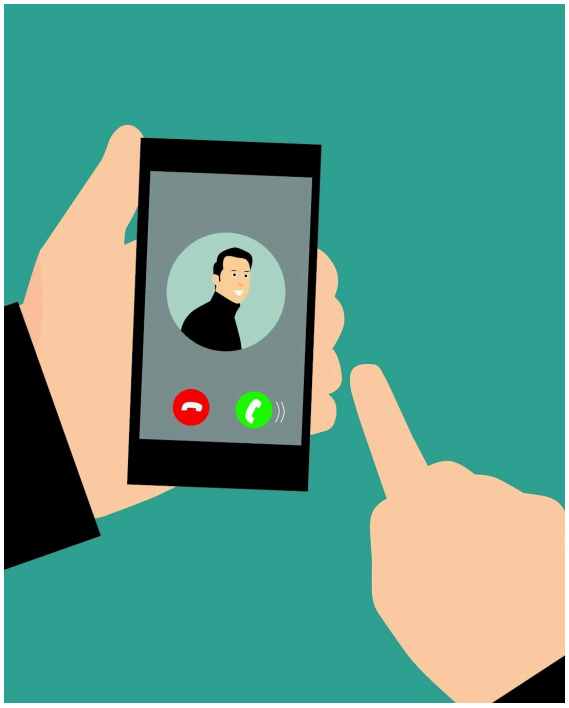
Solution 1

Used Firebase's real-time database and anonymous authentication to keep things fast and simple.

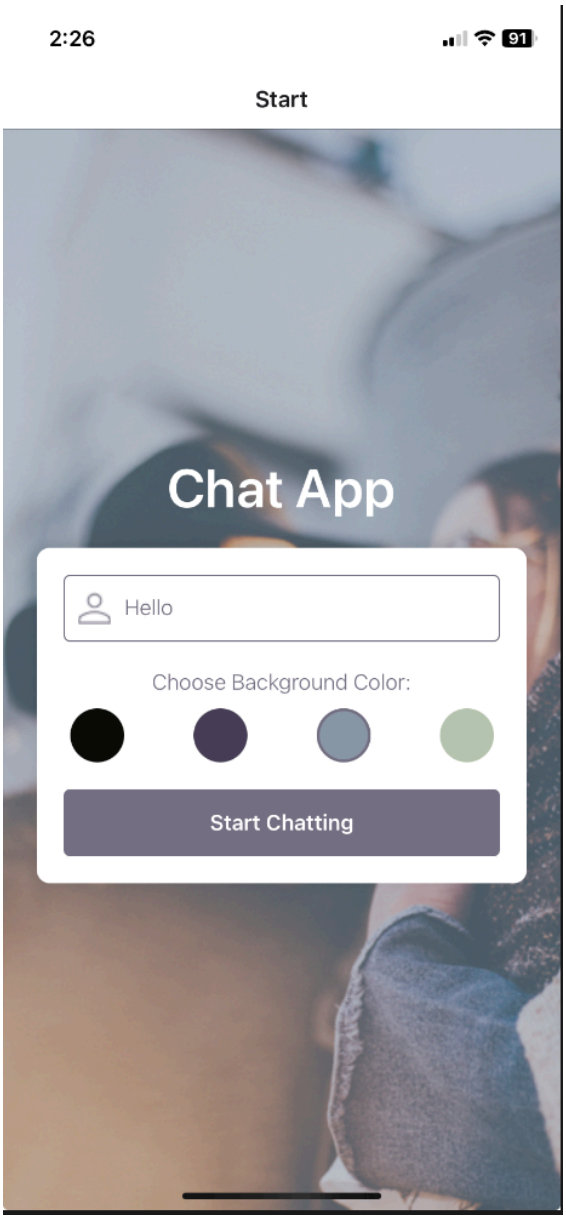
Solution 2

Integrated device tools like the camera, photo library, and GPS so users could share more than just messages.

Solution 3



Added offline message storage so users could continue chatting, even without service, and sync when back online.



Impact

Impact 1

Users had a smoother experience because they didn't have to create an account or remember a password.

Impact 2

The app provided a richer messaging experience with the ability to send images and locations.

Impact 3

I strengthened my mobile development skills, especially in real-time syncing, device permissions, and offline-first design.

Case Study Introduction

Background

Chat App is a mobile messaging application I built using React Native and Expo. It allows users to send text messages, share photos from their device or camera, and even send their current location.

The app works in real time using Firebase and also stores messages offline so users can keep chatting even without internet access.

I created this project to explore mobile development and learn how to build a fully functional, cross-platform app that feels simple and fun to use.

Objectives

Objective 1

Create a real-time mobile chat app that works on both Android and iOS devices

Objective 2

Allow users to send messages, share images, and send their location using built-in device tools

Objective 3

Support offline functionality by storing messages locally and syncing when reconnected

Objective 4

Provide simple personalization with a color picker to customize the chat screen

Problem

Most mobile chat apps require users to register or log in, which can slow down the experience. Some don't work well in low-connectivity environments or lack features like media and location sharing.

I wanted to build an app that anyone could open and use right away, even without an account, while still offering meaningful features that make conversations feel more engaging.

Recommendations

Recommendation 1

Start users with simple onboarding with no signup needed.

Recommendation 2

Support media and location sharing to make conversations more dynamic.

Recommendation 3

Build offline support so the app keeps working in areas with spotty service.

Impact of recommendation

SOLUTION	EFFECTIVENESS	IMPACT	NOTES
Anonymous login	Highly effective	Lowered the barrier for starting a conversation	Users can jump in quickly
Media and location sharing	Highly effective	Made messaging more fun and personal	Used built-in camera and GPS tools

SOLUTION	EFFECTIVENESS	IMPACT	NOTES
Offline message storage	Moderately effective	Improved reliability in all conditions	Messages are saved locally and synced later

Analysis

Research methods

- Studied messaging apps for must-have features
- Tested app on both iOS and Android devices
- Collected feedback from testers about layout, usability, and offline performance

Approaches used

- Built with React Native and Expo for cross-platform support
- Focused on user flow and reducing friction

- Followed mobile UI best practices with a simple and familiar chat layout

Relevant facts and information

- Duration: 3 weeks
- Tools: React Native, Firebase, Expo, Gifted Chat
- Final Features: Real-time messaging, media sharing, location sharing, offline support, anonymous login

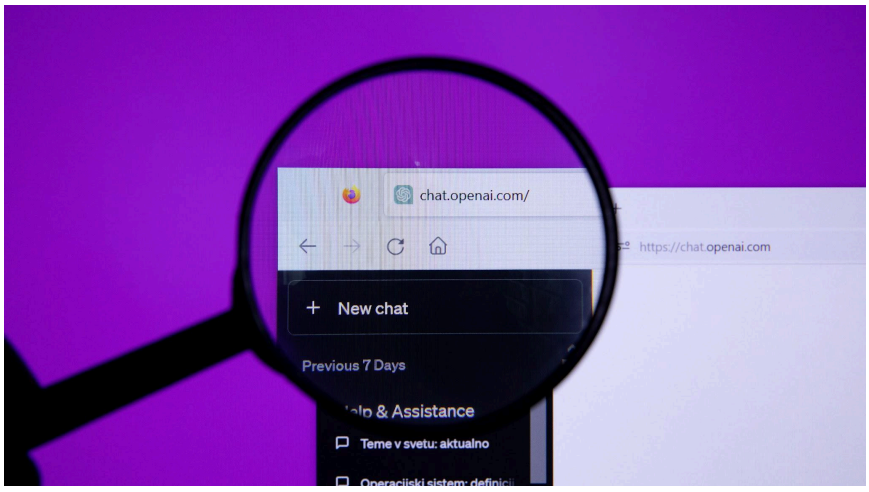
Challenges

One challenge was managing device permissions consistently across Android and iOS. For example, location and camera access worked differently between platforms.

Another challenge was setting up offline support so that messages saved locally would still sync properly once users went back online.

Working through these helped me understand how mobile apps handle real-world data syncing and background activity.

Conclusion



Summary of findings

The Chat App gave users an easy, flexible way to stay connected without requiring them to register or worry about losing connection. It offered just enough customization and interactivity to feel personal without becoming overwhelming.

Implications of the study

This project helped me grow as a developer by applying what I learned about mobile interfaces, real-time features, and cross-platform testing. I came away with a stronger understanding of how mobile tools like Firebase, camera access, and offline storage come together to build apps that feel smart and reliable.

> **“This project helped me build confidence in mobile development by creating something real people could pick up and use right away.”**