CHAT APP

case study

Theodoros Georgitsis

OVERVIEW

ChatApp is a mobile application, written in React Native in combination with the popular open-source library Gifted Chat. The app allows users to log in anonymously and chat with others in real-time. It also enables users to send pictures from their mobile device's library, access the camera to take new pictures, and share their locations. To install the app on a mobile phone, the Expo software was utilized.

See the project on Github

PURPOSE

I crafted this project as part of my full-stack web development course at CareerFoundry to showcase proficiency in React Native development.

OBJECTIVE

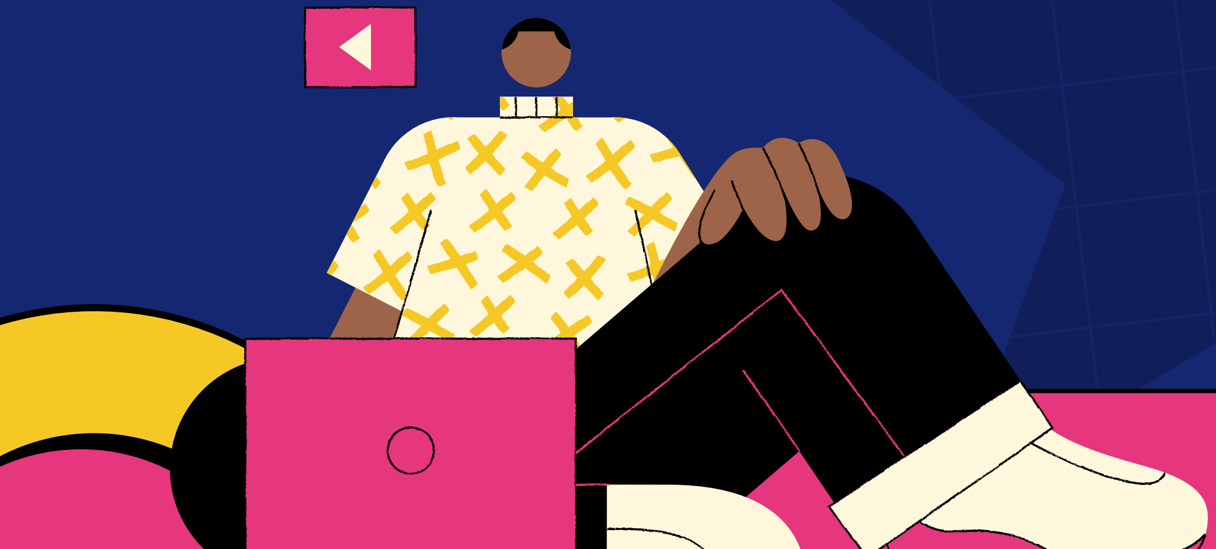
The objective of this project was to construct a web application using React Native on the client side and leverage a real-time database to store the messages. The completed project serves as a valuable addition to my professional portfolio.

METHODOLOGIES

- **React Native**
- **Gifted Chat**
- **Cloud Firestore**
- **Android Studio Emulator**
- **Expo**
- **Expo GO**

THE BACK-END

For ChatApp's back-end, I leveraged Google's Firebase. This entailed two primary configurations: enabling anonymous sign-in to assign and store unique user IDs, and integrating Cloud Firestore to efficiently manage message storage in a NoSQL database, making them accessible to all front-end users.



THE FRONT-END

For the Front-end of the Chat-App, I opted for React Native and the open-source library Gifted Chat. React Native's efficiency in cross-platform development, combined with Gifted Chat's robust components that simplify the implementation of real-time chat functionalities, not only contributes significantly to an enhanced user experience but also saves time and effort by eliminating the need to create every element of a chat app from scratch.

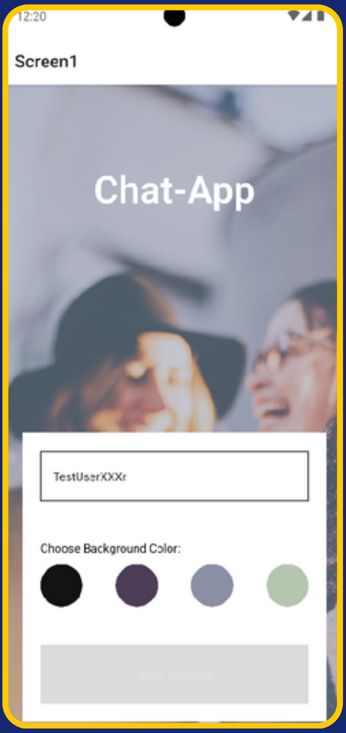
THE DEVELOPEMENT PROCESS

Android Studio Emulator and Expo GO where used to see the code created in action. Expo GO enabled me to install the Chat-App on different physical devices and Android Studio Emulator allowed me to emulate the usage of the front-end code on different android devices.

THE APP

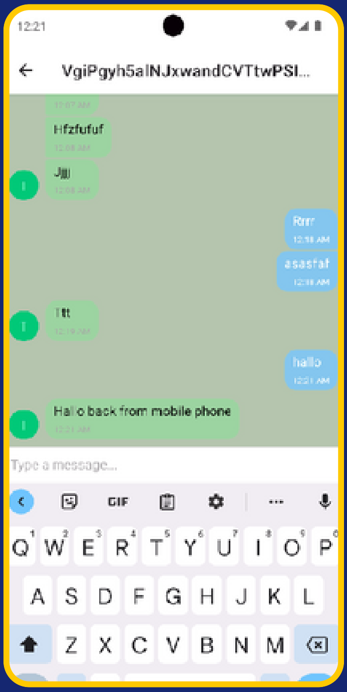
After the app has been installed the first view the user sees is the login view where a background color and user name can be chosen.

The chat view enables the user to send messages, share pictures, the users current location or voice messages.



TestUserXXXr

Choose Background Color:



Hfzfufuf

Jjjj

Rrrr

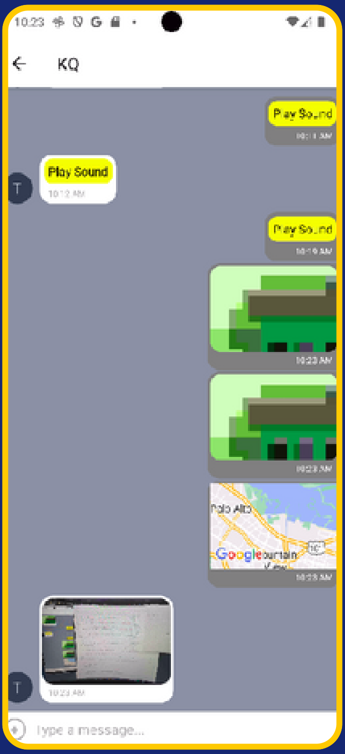
asasfaf

Tttt

hallo

Hallo back from mobile phone

Type a message...



Play Sound

Play Sound
10:12 AM

Play Sound

10:23 AM

10:23 AM

10:23 AM

10:23 AM

Type a message...

RETROSPECTIVE

What went well?

After the app has been installed the first view the user sees is the login view where a background color and user name can be chosen.
The chat view enables the user to send messages, share pictures or the users current location.

Lessons Learned for Future Projects

While navigating through various platforms and software, I've realized the importance of thorough documentation for any upcoming projects. Moving forward, I aim to enhance the documentation process to ensure better clarity and accessibility.

The Project Hiccup

The primary technical challenge I encountered during the development of the chat app was a networking error stemming from my wifi adapter, which proved to be less reliable than required for the WebSocket connection. This issue, unfamiliar to me until now, took some time to troubleshoot before realizing that there was no fault in the programmatic implementation.

Future Plans for the Chat App

At present, there are no immediate plans to introduce additional features to the app. However, my positive experience working with React Native has sparked an interest, and I look forward to embarking on my next React Native project in the near future.

CREDITS

Lead Developer:	Theodoros Georgitsis
Tutor:	Frank Harerimana
Mentors:	Tamim Abbas Aljuratli