

MARCH

2024

CREATED WITH REACT-NATIVE

KYLE MICHAELS

A FULL STACK CASE STUDY

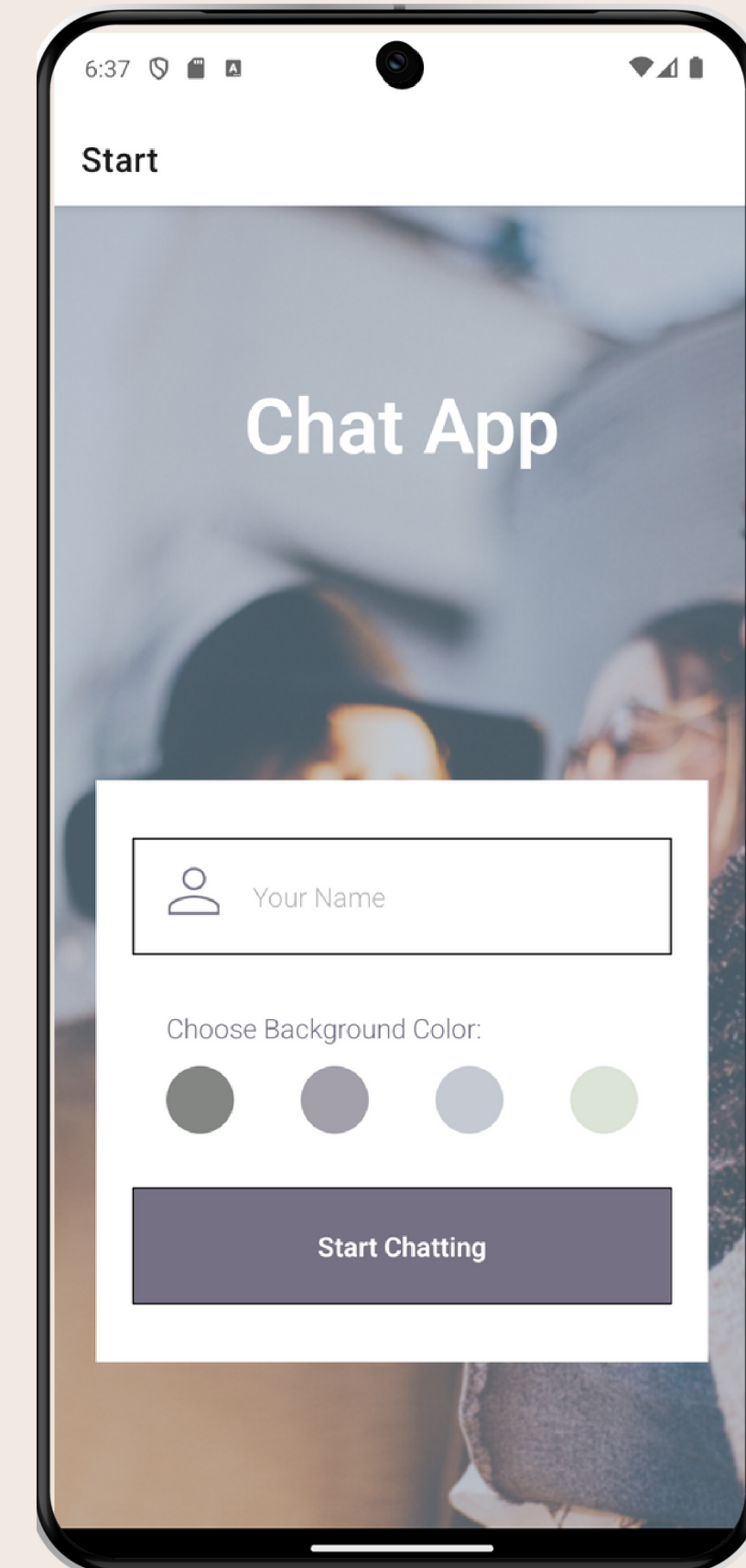
CHAT MOBILE APP

OVERVIEW

Chat is a mobile app built with React-Native that provides users a real time chat interface. Users are also able to share locations, images, and connect with others, all in one app.

CONTEXT

The Chat Mobile app is meant to showcase my abilities to create a mobile app from start to finish as a part of the CareerFoundry Full Stack Development course.



OBJECTIVES

To create a chat app that allows users to engage in real-time conversations through instant messaging.

To offer a user-friendly chat interface where users can exchange messages easily.

To provide users the functionality for sharing and uploading images, as well as accessing location services.

To showcase my skills in developing a React-Native app through the entirety of it's development cycle.



Time spent working on Chat via Ora.com

CHALLENGES

The biggest challenge I faced during this project was time. Since I am employed full-time, I had to balance learning new technologies such as Expo and Firebase while working with a deadline.

SOLUTIONS

In order to finish this project, I had to take a look at what was in front of me and make a game plan. After reviewing the documentation for Expo and Firebase, I was able to set aside time to grasp the concepts and apply them to my app. Along with gaining an understanding of my project, I had to schedule my time appropriately in order to succeed within my desired timeline.

TOOLS USED

React-Native

- Gifted Chat
- StackNavigator
- NetInfo
- AsyncStorage

Google Firebase

- Firestore Database
- Authentication
- Storage

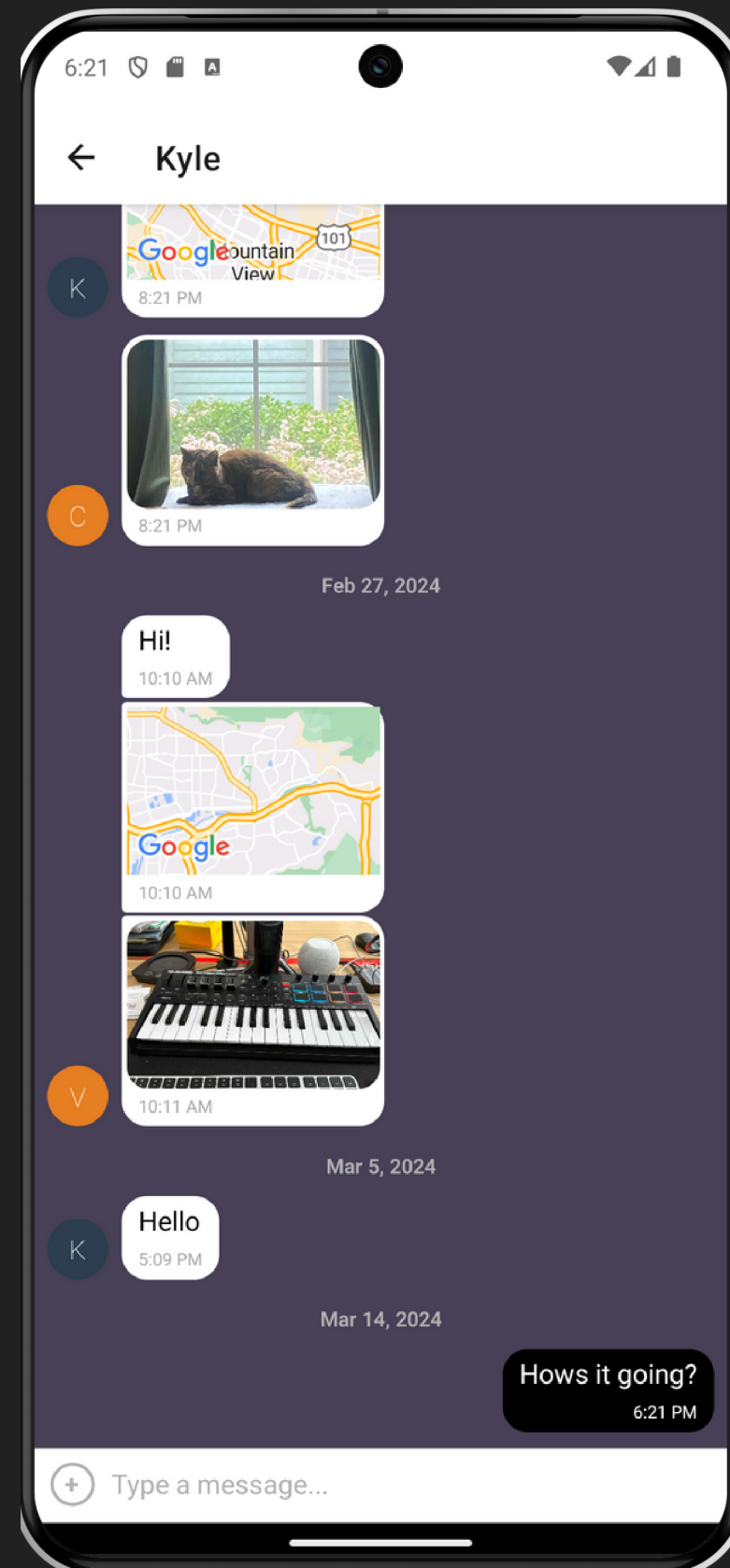
Expo

- Expo Go
- Expo SDK
- Expo CLI

JS Start.js JS CustomActions.js JS Chat.js X

VS Code > fullStackImmersion > 5NativeAppDevelopmentAndReactNative > chat > components > JS Chat.js > Chat

```
9  const Chat = ({ route, navigation, db, isConnected, storage }) => {
18    useEffect(() => {
31      unsubMessages = onSnapshot(q, (documentsSnapshot) => {
44        });
45        // if offline loads chached messages
46      } else loadCachedMessages();
47
48      //Clean up code
49      return () => {
50        if (unsubMessages) unsubMessages();
51      }
52
53      // Checks if isConnected changes. If true runs use effect function.
54    }, [isConnected]);
55
56
57    // Retrieves cached messages from storage and parses
58    const loadCachedMessages = async () => {
59      const cachedMessages = await AsyncStorage.getItem("chat_messages") || [];
60      setMessages(JSON.parse(cachedMessages));
61    }
62
63    // Stores messages from database when online in AsyncStorage
64    const cacheMessages = async (messagesToCache) => {
65      try {
```



SUMMARY

In order to finish this project, I had to take a look at what was in front of me and make a game plan. After reviewing the documentation of Expo and Google Firebase, I was able to set aside time to grasp the concepts and apply them to my app. Along with gaining an understanding of my project, I had to schedule my time appropriately in order to succeed within my desired timeline.

LESSONS LEARNED

- How to use React-Native to build a responsive real time messaging app.
- How to use and style GiftedChat components.
- How to create, access and write to a Google Firebase Database

DURATION

This project took a total of ten days to complete.

CREDITS

Lead:

Kyle Michaels

Tutor:

Francis Kim

Mentor:

Vinicius De Antoni