Backend – Achievement 5 – Native App Development & React Native

1) Description of the project

- What was my role for this project and what tasks did I face?
    - Building a JavaScript chat app built with React Native, featuring advanced functionality such as offline storage or geolocation

- Key Features
    - A page where users can enter their name and choose a background color for the chat screen before joining the chat.
    - A page displaying the conversation, as well as an input field and submit button.
    - The chat must provide users with two additional communication features:
        - sending images
        - and location data.
    - Data gets stored online and offline
    - Users can enter their name and choose a background color for the chat screen before joining the chat
    - Send and receive messages
    - Send and receive images (from media library or device's camera)
    - Send and receive locations
    - Record, send and receive audio
    - Users can view previous messages when offline

- Technical Requirements

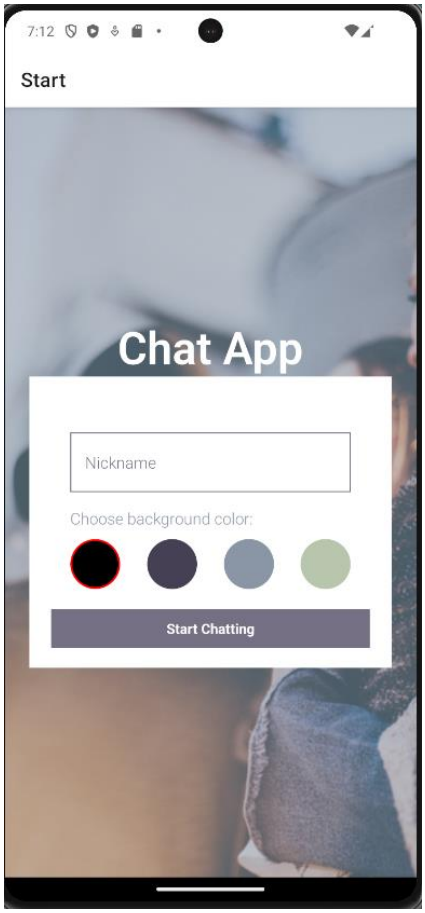
    - The app must be written in React Native.
    - The app must be developed using Expo.
    - The app must be styled according to the given screen design.
    - Chat conversations must be stored in Google Firestore Database.
    - The app must authenticate users anonymously via Google Firebase authentication.
    - Chat conversations must be stored locally.
    - The app must let users pick and send images from the phone’s image library.
    - The app must let users take pictures with the device’s camera app and send them.
    - The app must store images in Firebase Cloud Storage.
    - The app must be able to read the user’s location data.
    - Location data must be sent via the chat in a map view.
    - The chat interface and functionality must be created using the Gifted Chat library.
    - The app’s codebase must contain comments.

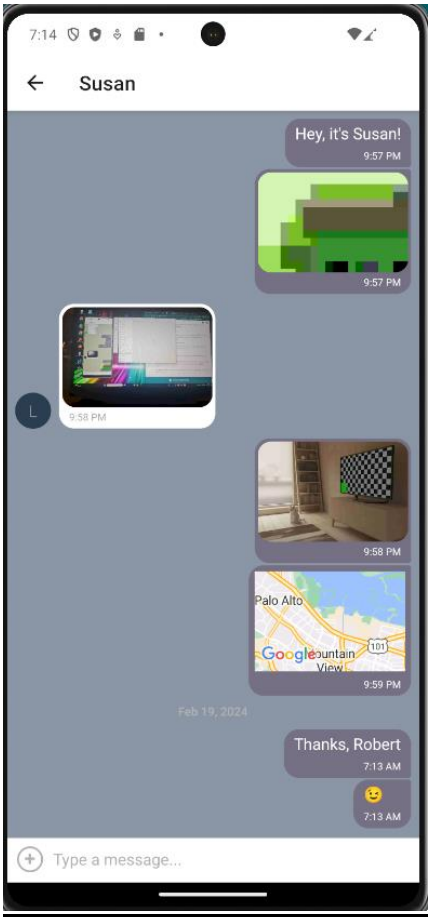
- Lessons I learned / decisions I made during this project

    - I articulated the distinction between native apps and other (web) app models
    - I established a framework, layout, and UI elements for a native app
    - I set up a virtual device to emulate your app
    - I implemented an accessible chat UI for a native app
    - I used and customized pre-build packages in a native app (Gifted Chat)
    - I learned customizing pre-build packages

- I addressed edge cases for different platforms and devices (iOS and Android)
- I compared different possibilities for implementing real-time apps
- I used Firestore to set up a database
- I managed to query a Firestore database in real time via a React Native app
- I rendered data from Firestore in a React Native app
- I implemented client-side data storage in React Native using AsyncStorage
- I described solutions for storing data on the client-side
- I learned how to partially enable some of the app's features when the device is offline using NetInfo
- I implemented communication features into a native app
- I implemented modern JavaScript APIs into an app
- I used and customize a third-party library
- Valuable lesson: Whether I'm developing a website, web app, or native mobile app, I must always think about how I'm handling users' data and respect their privacy—especially when working with a device's camera and microphone. Many users distrust products that use communication features, and it's part of a designer or developer's job to create apps that make the user feel secure. When users give my app access to their camera (or audio), I need to ensure that I don't breach their trust. For example, I must not save the image or video (or audio) files if it's not necessary for my app to do so.

2) A screenshot to represent the project





3) A link to the project's GitHub repository

- https://github.com/Luisa-Inc/chat-demo

4) A link to a video version of my app

- Chat App Demo Video

5) A list of the technologies used for each project

  - React Native: Framework for building mobile applications using JavaScript and React
  - Expo: Development platform for building React Native applications.
  - GiftedChat: A library for creating chat interfaces in React Native applications.
  - Google Firebase: Cloud-based platform that provides various services, including Firestore for real-time database and authentication.
  - AsyncStorage: Local storage system in React Native for caching and persisting data.
  - Expo ImagePicker: Expo API for accessing the device's image picker to choose images from the gallery.
  - Expo MediaLibrary: Expo API for accessing and managing media assets on the device.
  - Expo Location: Expo API for obtaining location information from a device.
  - react-native-maps: React Native Map components for iOS + Android.
  - MapView: Specific component from the react-native-maps library used to display maps in React Native applications.

6) Any other relevant materials I created for the project

- User Stories
  - As a new user, I want to be able to easily enter a chat room so I can quickly start talking to my friends and family.
  - As a user, I want to be able to send messages to my friends and family members to exchange the latest news.
  - As a user, I want to send images to my friends to show them what I'm currently doing.
  - As a user, I want to share my location with my friends to show them where I am.
  - As a user, I want to be able to read my messages offline so I can reread conversations at any time.
  - As a user with a visual impairment, I want to use a chat app that is compatible with a screen reader so that I can engage with a chat interface.

- Design Assets
  - Can be found here.