Project 01
Android based API app

Description

For this project teams will work together to build an app that consumes information from a REST API and uses a local database for caching. The values from the API may be stored in the local database.

The app will be built using React Native.

Example apps:

Job listings

User creates a login and selects a set of interests from a drop down. The app uses a job API list to list jobs based on the skills selected. Maybe let the user save the jobs to the phone (I'd say save the ID to the ROOM DB and load it when the app loads). Users should be able to modify their profile to change what jobs they are interested in.

Fitness App

Users create a login and can then select exercises provided by a fitness API. Username and password should be used to get an API token that will be stored locally. The app will allow the user to store exercises and workout plans.

Sports Win Predictor

Users create a login and select their favorite teams. The app then gives the user predicted outcomes of upcoming matches. It will also allow them to change which teams they want to follow. The basis of the win is that, across all sports, teams that play at home have a > 50% chance of winning.

Learning Outcomes Addressed

1. Apply, Analyze, and Integrate software engineering processes.
   Including:
   a. requirements analysis
   b. Software design
   c. test driven development
   d. code review
   e. continuous integration and deployment
   f. agile planning and execution (i.e. user stories)
   g. retrospectives
   h. peer feedback.
2. Practice and Evaluate Software Development lifecycles
   a. Specifically: Continuous integration, Incremental development, waterfall development and others as time permits
3. Build and Evaluate software using the agile development model as defined in the textbook used in this course.
4. Evaluate, Explain, and Apply team management skills
   a. Establish clear team roles and expectations
   b. Set realistic milestones
   c. Peer-review and evaluation
   d. Effective communication
   e. Project management
5. Identify, Explain, and Apply proper source control
   a. Manage commits to a shared repository
   b. Use proper comments and style when adding code to git
6. Critique, Prepare, and Evaluate Professional Communication
   a. Prepare and present information about projects
   b. Critique and assess other student work, including presentations.

Resources

- Getting started with Expo/React
  - https://docs.expo.dev/tutorial/create-your-first-app/
  -
- Consuming an API with React Native
  - https://reactnative.dev/docs/network
  - Axios and React Native
  - https://medium.com/enappd/how-to-make-api-calls-in-react-native-apps-eab083186611
- Fake API for mocks and testing: jsonplaceholder.typicode.com/
- A list of free APIs: https://github.com/public-apis/public-apis
  - Pixel Avatars: https://avatars.dicebear.com/

Minimum Viable Product (MVP)

- □ Fetch data from an API
- □ Store information locally
- □ Display some information
- □ Authenticate users
    - □ Look in to OAuth though it is not required
- □ (optional ish) Create users

Must haves (non-negotiable)

- MUST pull from a web based REST API
    - See here for a list
    - Import Yeti
- MUST use React Native/Expo
- MUST use local storage
    - https://github.com/andpor/react-native-sqlite-storage
    - https://www.youtube.com/watch?v=GoDERit8mVo
    - 100% inflexible on this. I don't want data stored on Firebase.
- MUST use Git and Github
    - Each week I will check for a pull request
    - The individual contributions to the github repo will be a major part of the evaluation of this project.
        - This means everyone MUST contribute. I don't want to hear "I didn't do anything because…"

Updates

Jan 14, 2025 V 2.5.1

- Formatting and cleaning up language
- Removed one requirement

Sep 6, 2024 V 2.5.0.

- Updated the language to not focus on the android Room library and instead to focus on local (SQLite) storage.

Aug 23, 2024 V 2.0.0.

- Added an update section
- Added versions