Assignment 4: Spotify 2.0

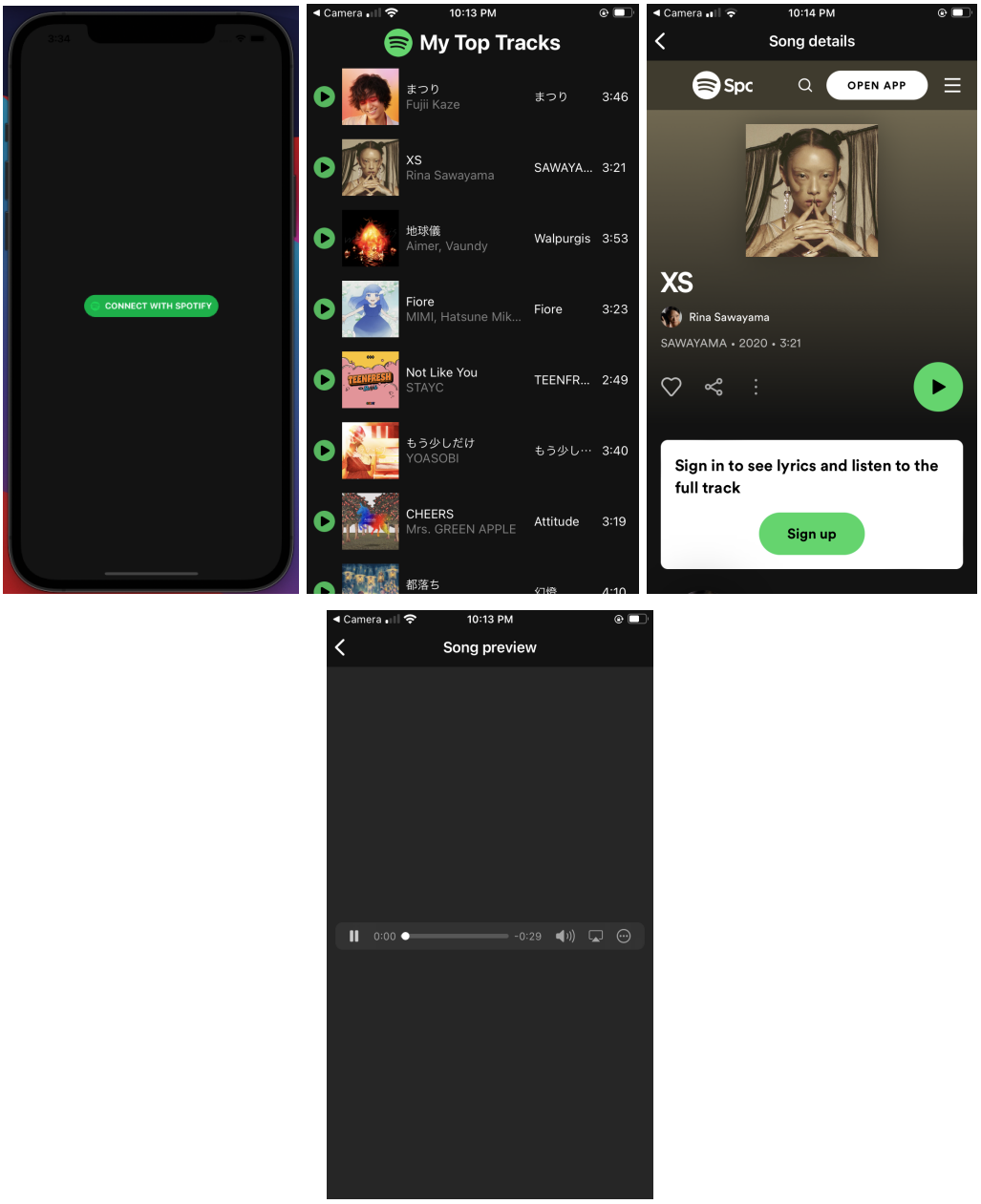
Due Monday, Nov 13th @ 11:59pm

Overview

After implementing a simple Spotify app, you found there to be a huge demand for such a music app. You meet up with your design team and you decide that you want to beef up your app and add more functionality. Together, your team decides to add playback capability and a new screen that shows more about the selected song. Since you have already built most of the app, your job is now to utilize a StackNavigator to turn your app into a single functional app with multiple screens (a song list screen i.e. the one you made in assignment 3, a more detailed song viewing screen given by the Spotify API, and a playback screen which is also given by the Spotify API). When we say the screen is given by the API, we mean that the Spotify API already gives us the URL for the page, and we just have to load that into a new screen! (i.e. the content show in the 3rd and 4th screenshots below are given to us, we won't have to manually create them :))

Design Details

Your final product will look something like this:



1 - Spotify Auth (part of main screen)
2 - List of Songs (part of main screen)
3 - Detailed Song screen
4 - Song preview screen

Since you have already built a Spotify app for Assignment 3, you will use your Assignment 3 code as a starting point for this project.

For this assignment, you will not need to focus much on the visual elements of the app. We'll be grading based on your implementation of the navigation screens and the navigation-related UI. Here's a breakdown of what we want to see:

Feature	Description
Visual Requirements	• Lean towards matching the content in the screenshots • Note: we replaced the song index display on a song with a play button!
Navigation bar	• You must use a Stack navigator to go from the main screen to the specific song viewing screen (which appears once you tap on a song row) and to the song preview screen (which appears once you tap the play button next to the song in the FlatList). Your navigation bar will show up automatically, but we want to hide it on this main screen (screenshots 1 & 2) but show it on the detailed song screen (screenshot 3) and song preview screen (screenshot 4). • The navigation bar should be a dark color that fits the app's color scheme and display the appropriate title ("Song details" or "Song preview"). • The text in the OS status bar (the top bar that displays the time, battery, etc) should be legible, i.e. in a light-colored font.
Detailed song screen and song preview screen	• The detailed song screen and song preview screen only has a single `WebView` component which will open a specific webpage given a URL. (This component is given to you through a library called react-native-webview)
Tap to play or view	• Tapping on **a song's row** should navigate the user to the detailed song screen and pass over the song's **external_url** to be displayed in a WebView. • Tapping on **the play button** next to the song in the list should navigate the user to the song preview screen and pass over the song's **preview_url** to be displayed in a WebView.

Implementation

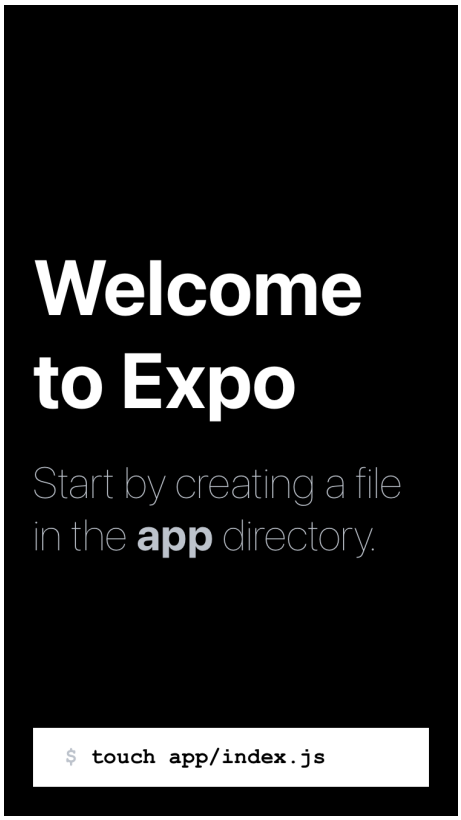
There are five main pieces to this assignment: (1) set up your Spotify API key, (2) implement the authentication hook `useSpotifyAuth`, (3) create the "Connect with Spotify" button, (4) implement the track fetching hook `useSpotifyTracks`, and (5) display a list of tracks.

Expo Router Setup

For this assignment, we will be building on top of your existing Assignment 3 implementation. To get Expo Router working with your codebase, we'll need to do some setup.

1) Open your terminal and navigate to the folder containing your Assignment 3 project.
2) Update your Expo version to 49: `npm install expo@^49.0.16`
3) Update all your project's dependencies for the new Expo version: `npx expo install --fix`ur
4) Follow the following **Manual Installation** steps in the Expo Router install guide: https://docs.expo.dev/routing/installation/#manual-installation
    a) Install dependencies
    b) Setup entry point
    c) Modify project configuration
    d) Modify babel.config.js

Once you've finished the above steps, your app should look like this:



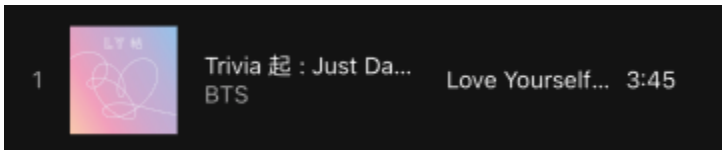
5) Make a new folder called `app/` , move App.js into the new folder, and rename it to `index.js` . This file, `app/index.js` , wil be the new home screen for your app.
6) Since all the imports in App.js were written relative to its path, you may need to update all the import statements appropriately (if your IDE doesn't handle it for you automatically).

7) Once you've finished the above steps, your app should be working again!

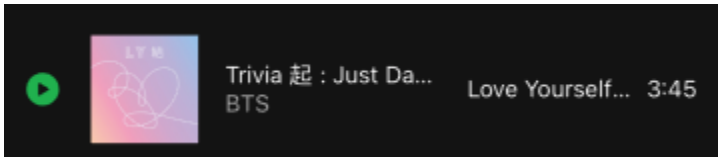
Play Buttons

Instead of a song index, we want to add a play button to each song. You should use an existing icon to build this button. https://docs.expo.dev/guides/icons/ is a great resource for Icons.

Before:



After:



Navigation and WebViews

You will need to implement navigation to go between THREE screens: the main screen (what you implemented in Assignment 3, i.e. the authentication button and song list), the detailed song screen, and the sound preview screen. When the user taps on a song, your app should open a new screen for showing details about the song. When the user taps on the play button, your app should open a new screen that gives a sound preview of the song.

For our purposes, opening the song's **external_url** property and **preview_url** property for the detailed song screen and the sound preview screen respectively (both found in the song object) in a WebView is enough. Your task is to figure out how to implement a WebView (see docs), set up a Stack navigator (see docs), navigate between pages (see docs), and pass params to the route that renders the WebView (see docs). A full Expo Router navigation setup is also provided in the code example from Lecture 11.

Polishing Navigation UI

Once you've implemented all the navigation functionality, you should take some time to make sure that it all looks polished. Here are specific things we'll be looking for:

- The navigation bar should only show up on the Song Details and Song Preview screens. It should NOT appear on the home screen.
- The navigation bar should be a dark color that fits the app's color scheme.

- The navigation bar should display the appropriate title for the screen (“Song details” or “Song preview”).
- The text in the OS status bar (the top bar that displays the time, battery, etc) should be legible, i.e. in a light-colored font. You can set this color using the setStatusBarStyle function provided by expo-status-bar.

Extensions

There are no extensions for this project. We encourage you to spend your time working on the final project instead!

Starter Files

Your A3 implementation is your starter code—there is no new code to download.

Deliverables

There are two deliverables for this assignment:

1) Submit a .zip file on Canvas containing the following:
    - The source code for your submission (include all files EXCEPT for the `node_modules` folder)
        - **Important:** Please delete the `node_modules` directory from your project. This ensures that your .zip file remains a manageable size and makes it much easier for us to download and grade your assignment. If you need to rerun expo or make any changes afterward, you can run `npm install` to reinstall the necessary modules. We may deduct points for neglecting to delete `node_modules`.
    - A screenshot of your app, as well as any additional screenshots documenting your extensions
2) Fill out this feedback survey: http://hci.st/cs147L-A4-survey

Grading

To receive a check-plus, we expect the following:

- Implement all the specs described in the “Design Details” section.

Your grade will be determined as follows:

- ✓++ Astounding; app is highly polished and implemented to spec with minimal or no errors
- ✓+ Great; satisfied all the design specs with minor errors at most

- ✓ Good; satisfied all the design specs but with some major or minor issues
- ✓- Okay; satisfied most of the design specs
- ✓- - Poor; satisfied some of the design specs but missing key aspects
- – Incomplete work or missing