CS624 Full-Stack Development – Mobile App
HOS01A: Cloud Development Environment
Reviewed by Anahita Raeiszadeh on September 20, 2023
Reviewed by Jonathan Koerber March 21, 2024
Reviewed by Clark Ngo February 27, 2024
School of Technology & Computing (STC)
City University of Seattle (CityU)

Overview:
React Native is a framework for building mobile applications using JavaScript and React. In this guide, we will walk through the process of setting up a React Native development environment with GitHub Codespaces, Expo, and Expo Go. The guide includes step-by-step instructions on how to set up GitHub Codespaces, install Expo Go, and create a basic React Native app. This guide is designed for students who are new to React Native development and want to learn how to use Expo Go and GitHub Codespaces to create mobile applications.

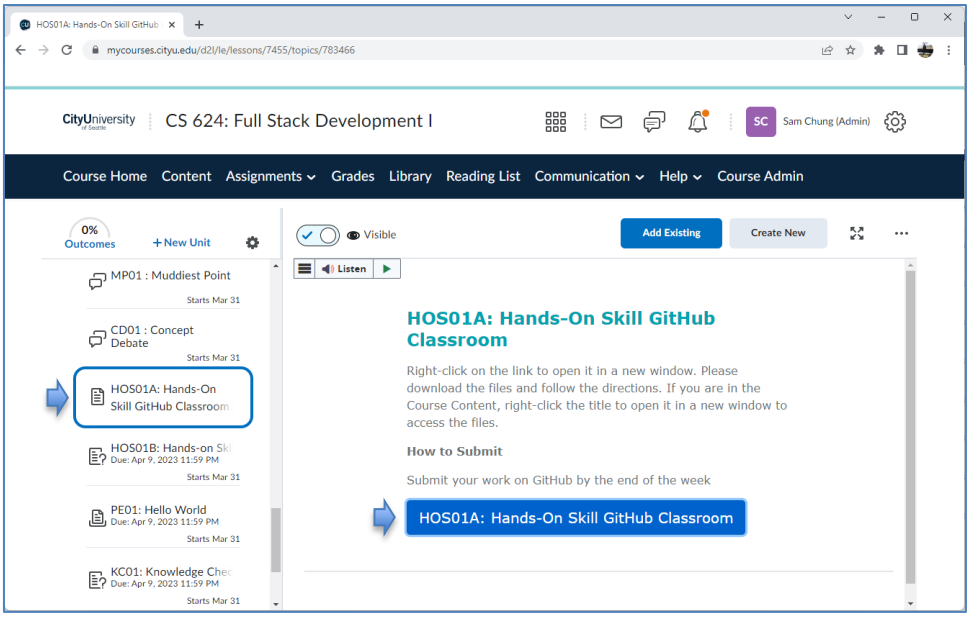
Before You Start
- Screenshots may be different from your environment.
- The directory path shown in screenshots may be different from yours.
- There might be subtle discrepancies along with the steps. Please use your best judgment while going through this cookbook-style tutorial to complete each step.
- Some steps may not be explained in detail. If you are not sure what to do:
  1. Consult the resources from the course.
  2. If you cannot solve the problem after a few tries (usually 15 -30 minutes), ask a TA for help.

Readings and Examples:
- Visit CS 624 Repository for Examples.
  - Select the related module.
  - Visit the README.md file.
  - Find examples for your practices.

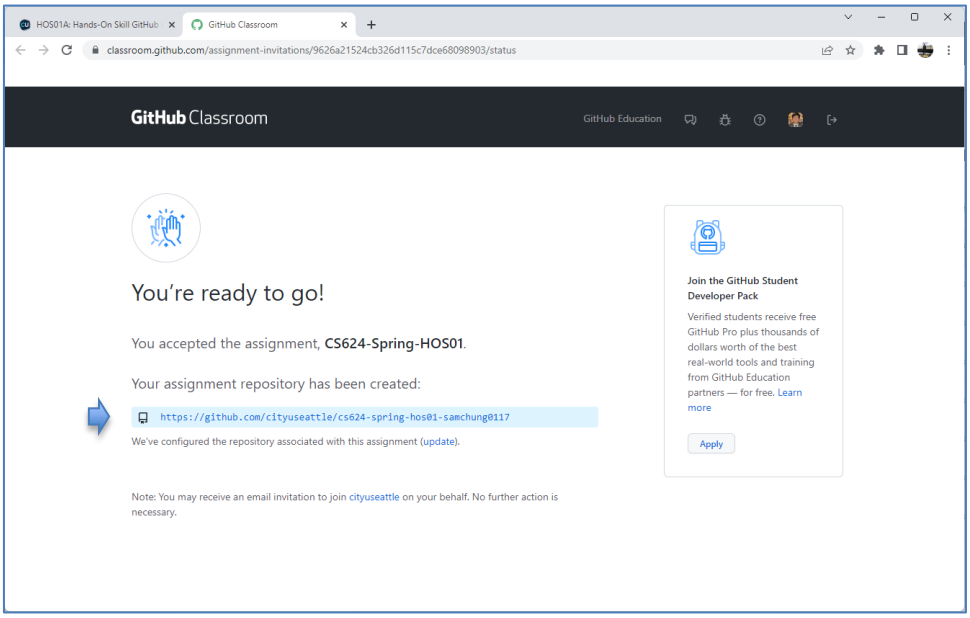
Learning Outcomes
- Section 1: Accessing GitHub Codespaces
- Section 2: Setting Up Expo account and Downloading Expo Go on my Smartphone
- Section 3: Creating my first mobile app
- Section 4: Updating my first mobile app
- Section 5: Pushing your work to GitHub

Section 1: Accessing GitHub Codespaces
GitHub Codespaces are online cloud-based development environments that allow you to easily write, run, and debug your code. It is fully integrated with your GitHub repository and provides a seamless experience for developers. It would help if you had a GitHub account and an active internet connection to access the Codespaces environment.

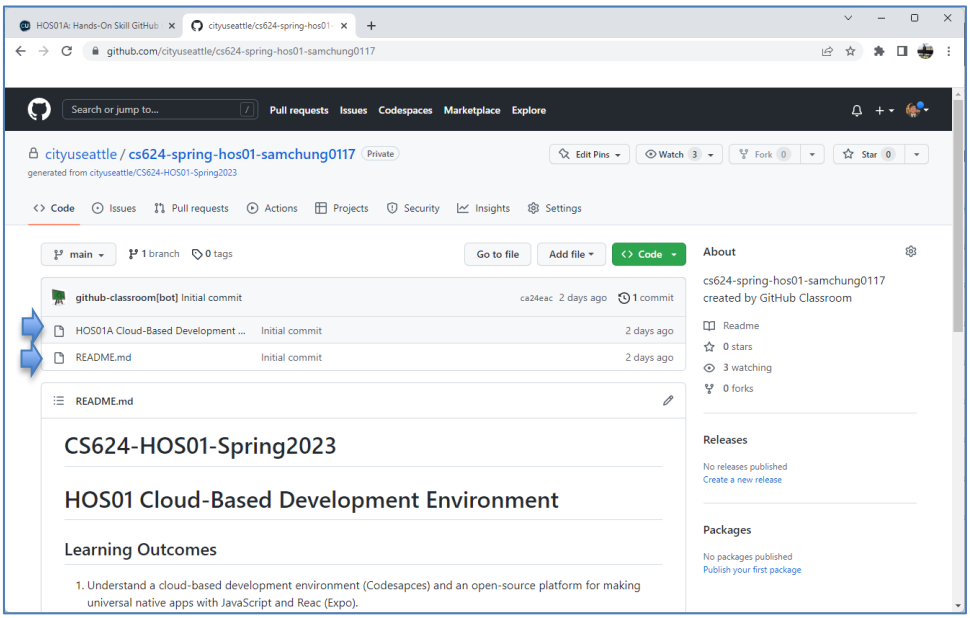
1) From your course shell, visit the HOS.
   Visit the HOS GitHub Classroom.



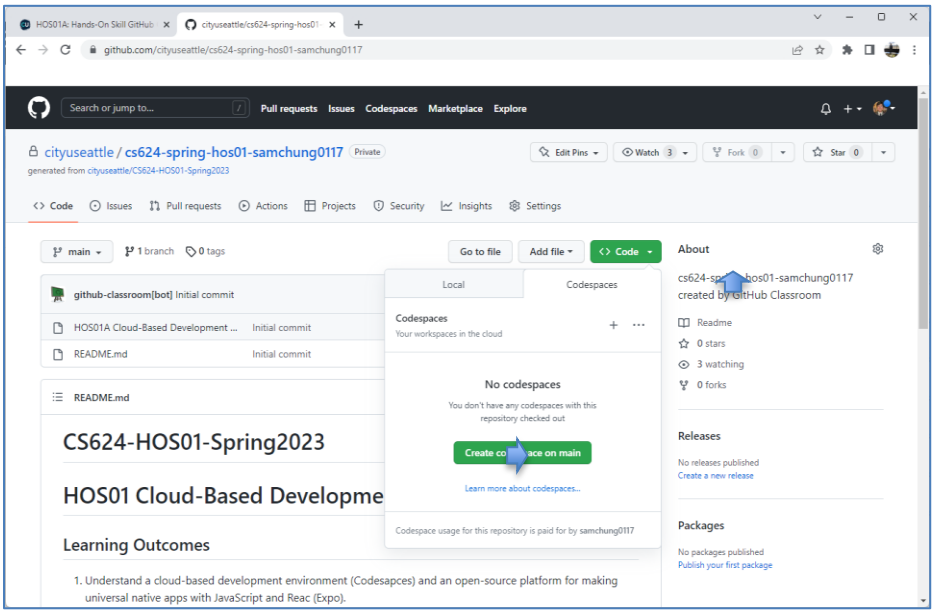
2) Go to your repository created from Brightspace on the GitHub classroom.



3) Download the given HOS guidelines that you will use for your hands-on skills.
Also, check the README.md for your HOS.

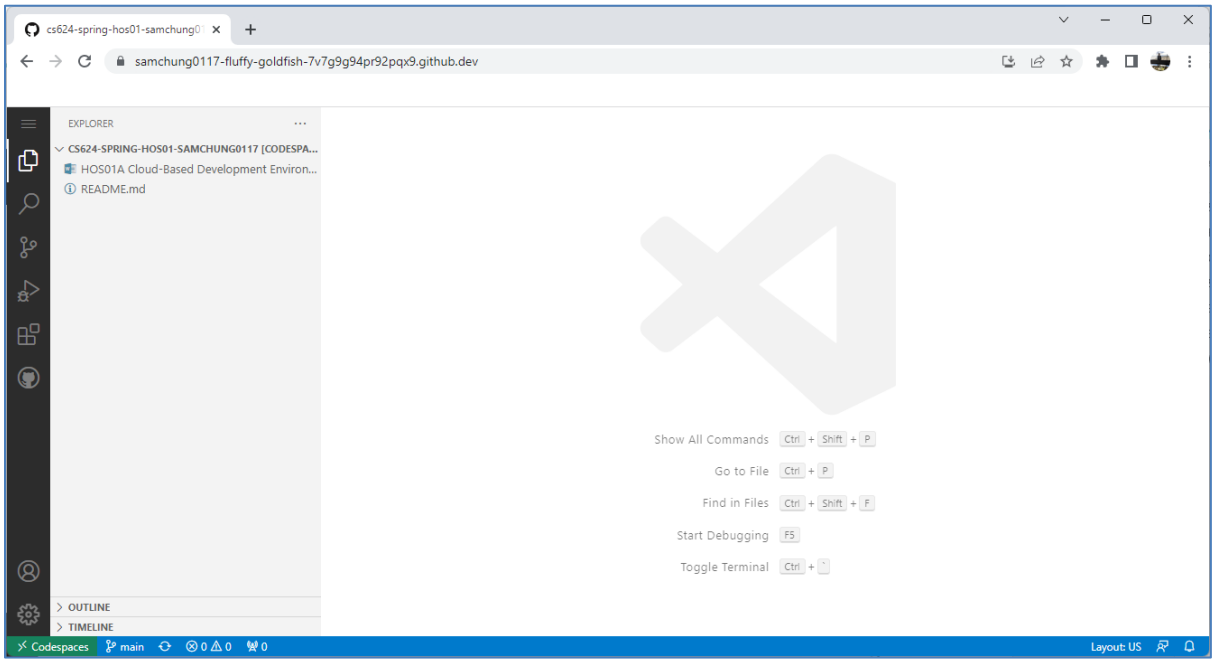


4) In the top-right corner, click on the “<>Code” drop-down menu.
Select "Create a codespace on main."



5) Wait for the Codespaces environment to load. Once loaded, you can access the terminal, file explorer, and other tools to start working on your project.

   Once your codespace is created, the template repository will be automatically cloned into it. Now you can run the application and launch it in a browser which we call cloud IDE (Integrated Development Environment). We use Visual Studio Code for the web.



6) What is the purpose of this section? i.e., why did you practice this section?

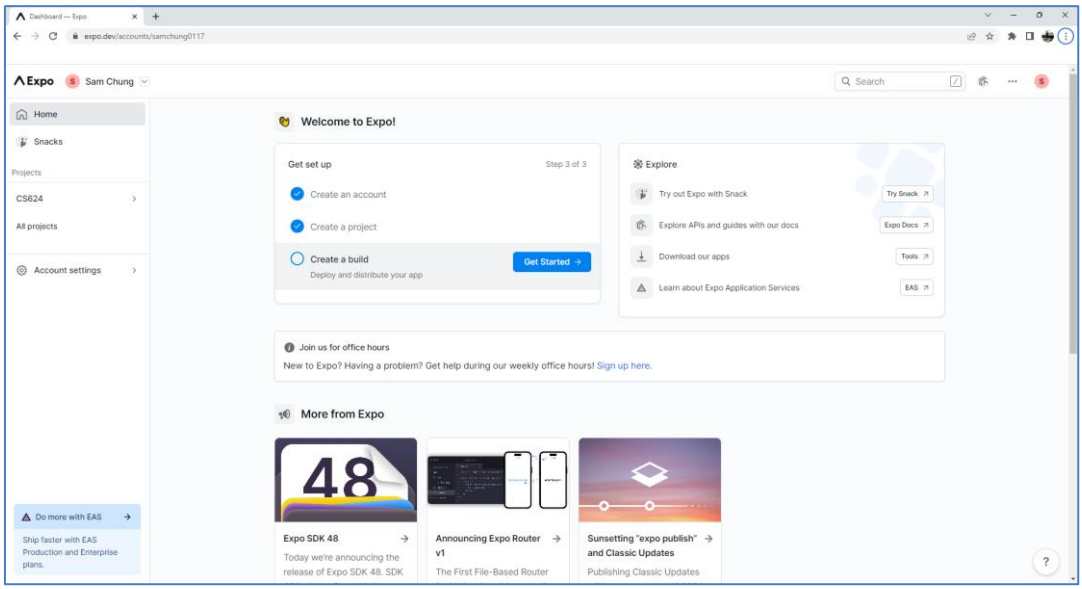
Section 2: Setting Up Expo environment

1) Creating my Expo account

"Expo is an open-source platform for making universal native apps for Android, iOS, and the web with JavaScript and React." (https://docs.expo.dev/introduction/faq/)

If you do not have an Expo account, sign up for an account in Expo here.
Remember the credentials. We need them for all the exercises.

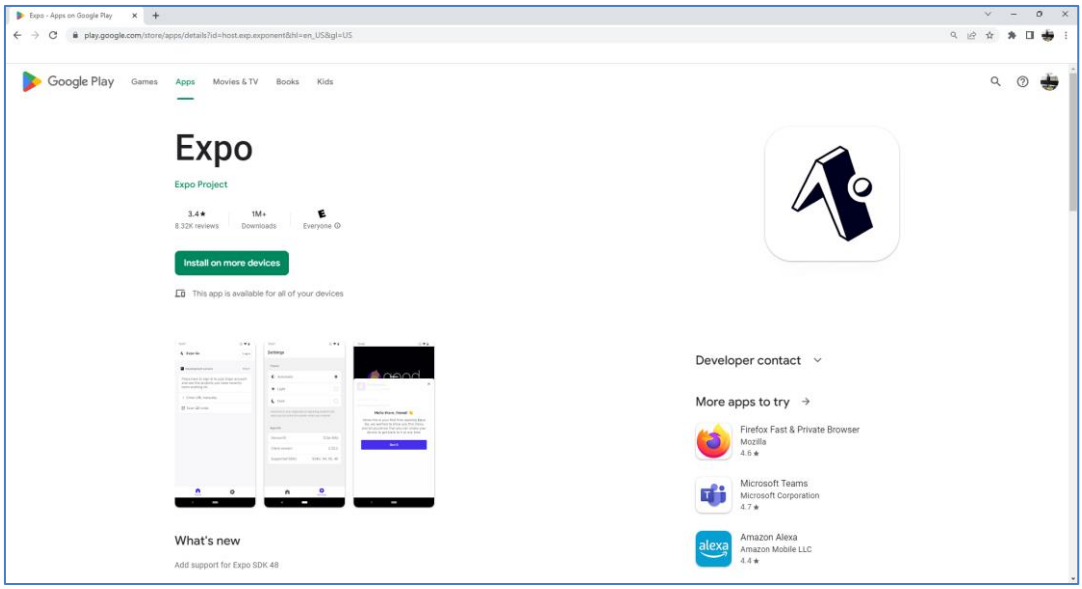
If you log into your Expo account, you can see a screen like the one below.



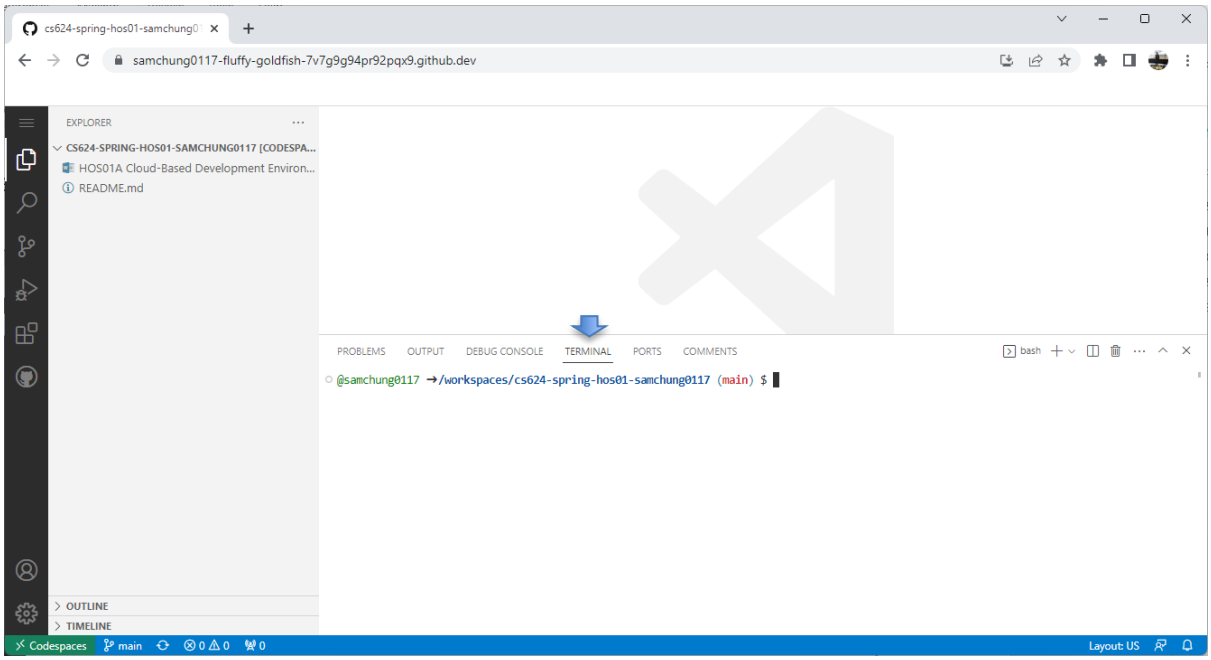
2) Downloading Expo Go on my smartphone

"Expo Go is a free, open-source client for testing React Native apps on Android and iOS without needing to build anything locally." (https://docs.expo.dev/workflow/expo-go/)

Download the "Expo Go" app from the App Store or Google Play Store.



3) Open the repository in GitHub Codespaces.
Open the terminal (press “~” and “ctl” together.)



4) Test the Node environment – node, npm, and npx
Assume that you are under the current working directory for HOSs.

“By default, the container for your codespace has many languages and runtimes, including Node.js, JavaScript, and Typescript. It also includes a common set of tools, such as nvm, npm, yarn, git, wget, rsync, openssh, and nano.” (https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration/setting-up-your-nodejs-project-for-codespaces)

“Node or Node.js is an asynchronous, event-driven, open-source, cross-platform JavaScript runtime environment.” (https://nodejs.org/en/)

“The command npm is used to download JavaScript packages from Node Package Manager, and npx is used to execute JavaScript packages downloaded this way.” (https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/)

By typing the following version check commands, you can confirm that your Codesapces have node, npm, and npx.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS        bash
@JonathanKoerber ➜ /workspaces/cs624-spring-2024-hos01-JonathanKoerber (main) $ node --version
v20.11.1
@JonathanKoerber ➜ /workspaces/cs624-spring-2024-hos01-JonathanKoerber (main) $ npm --version
10.2.4
@JonathanKoerber ➜ /workspaces/cs624-spring-2024-hos01-JonathanKoerber (main) $ npx --version
10.2.4
@JonathanKoerber ➜ /workspaces/cs624-spring-2024-hos01-JonathanKoerber (main) $
```

5) Install Expo CLI (Command Line Interface)
"Expo CLI (Command Line Interface) is a command-line tool that is the primary interface between a developer and other Expo tools." (https://docs.expo.dev/workflow/expo-cli/)

Install Expo CLI globally.
In the terminal, run the command **npx expo login.** If Expo is not installed, you will be asked if you would like to install expo. BASH CLI tools usually have some help function to see available commands by running **npx expo --help.** You can clear the terminal output by running **clear** command.

```
@JonathanKoerber ➜ /workspaces/cs624-spring-2024-hos01-JonathanKoerber (main) $ npx expo
-h

  Usage
    $ npx expo <command>

  Commands
    start, export
    run:ios, run:android, prebuild
    install, customize, config
    login, logout, whoami, register

  Options
    --version, -v   Version number
    --help, -h      Usage info

  For more info run a command with the --help flag
    $ npx expo start --help
```

6) Next in the terminal, log into Expo using the command **npx expo login**.
When prompted, enter your credentials if you did not log in already.

7) What is the purpose of this section? i.e., why did you practice this section?

Section 3: Creating my first mobile app

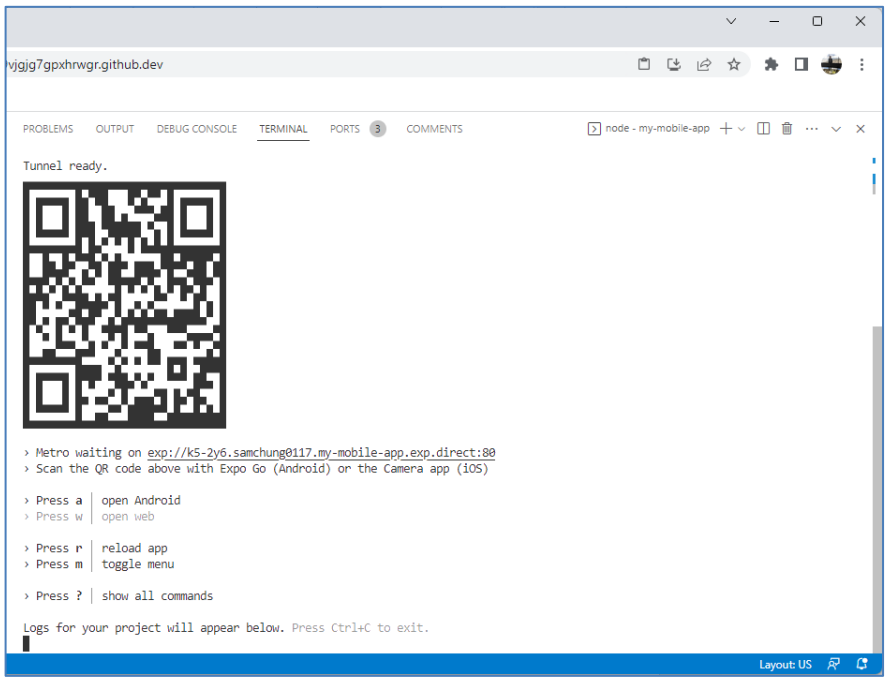
1) Lets check to see that we are in our root directory by typing into the terminal prompt `**pwd**` and pressing enter. Your root dir should be /workspaces/cs624-spring-2024-hos01-<your github username>. We are going to use the command line tool `**create-expo-app**` to build our app. Let's have a quick look at the available commands that by running `**npx create-expo-app --help**` as you can see this has a lot of functionality.

   Run the command **npx create-expo-app my-mobile-app** to create a new Expo project.

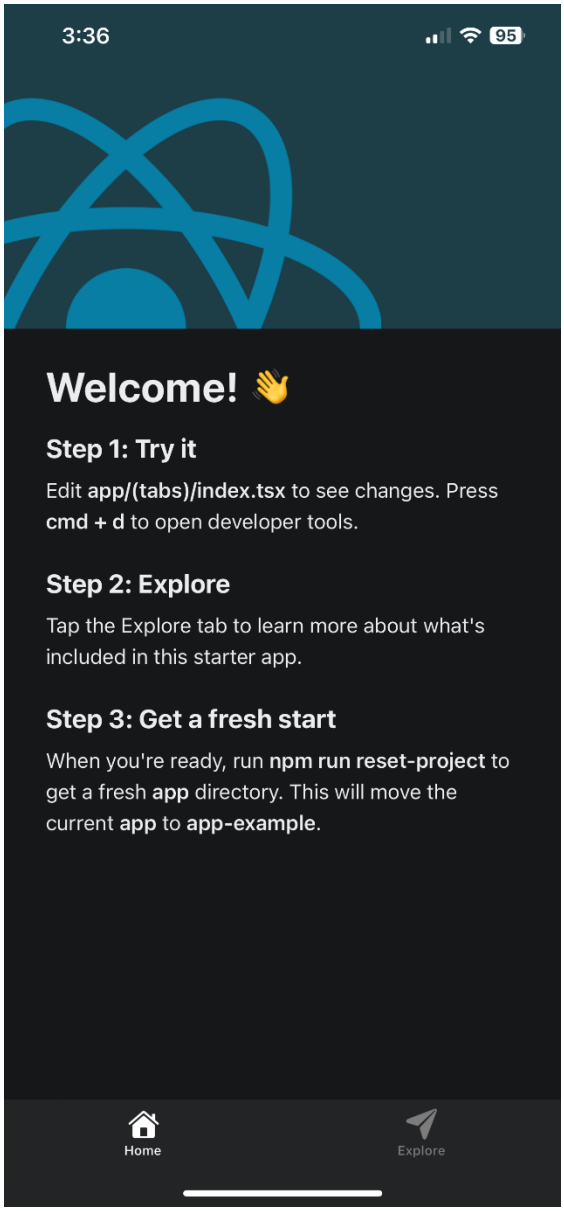
2) Next run `**ls**` to see what in our root directory. You can see a directory called “my-mobile-app” was created.

3) Change the current directory to the newly created project using the command **cd my-mobile-app**.
4) Run the command **npx expo start --tunnel** to start the development server.

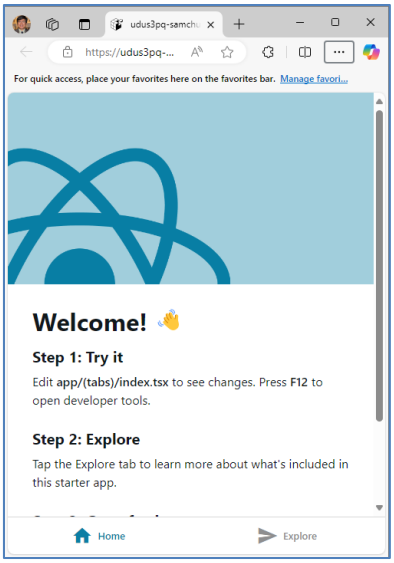
5) Preview the app in the Expo Go client app on your mobile device by scanning the QR code with Expo Go App (Android) or the Camera app (iOS) or accessing it thought Expo Go app.



Each smartphone provides a feature to take a screenshot.
The captured screenshot is shown below.



6) Play with the bottom menus “Explore” and “Home.”
7) Use ctrl + C to stop the server.
8) Close the Expo Go app on your mobile device.
9) Run the command “**npx expo start --tunnel”** to start the development server.
10) This time, press “w” to open the web version – “Press w | open web”

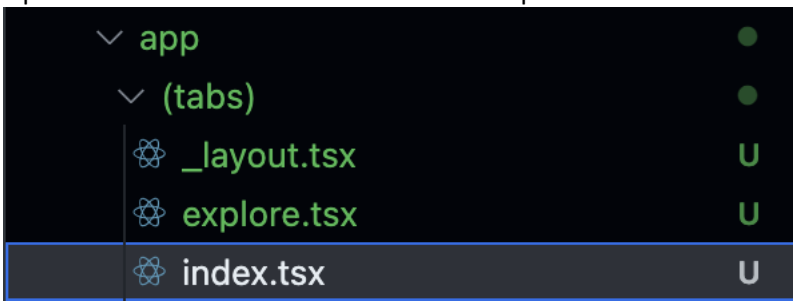


11) Play with the bottom menus “Explore” and “Home.”
12) Use ctrl + C to stop the server.
13) Close the web browser.
14) What is the purpose of this section? i.e., why did you practice this section?

Section 4: Updating my first mobile app

Steps to change the background color of the app:

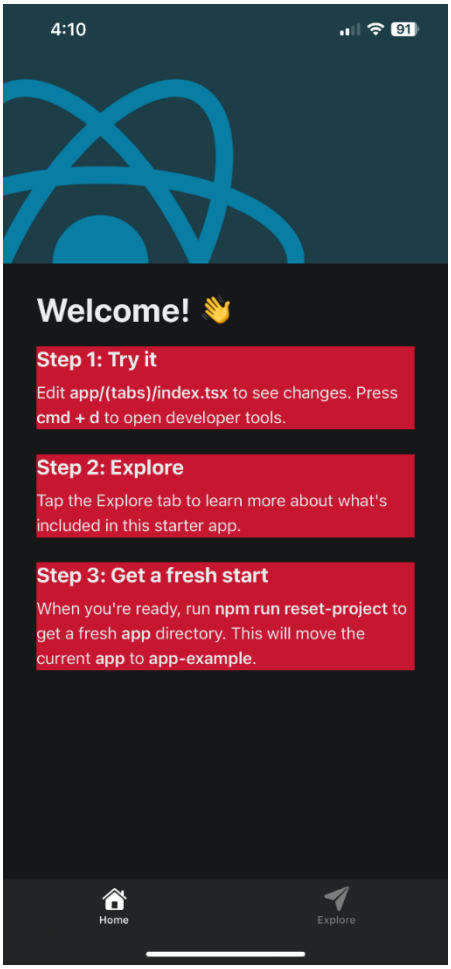
1) Open the **index.tsx** file in the GitHub Codespaces environment. Located under apps/(tabs)/



2) In the **styles** object, locate the **stepContainer** property.
3) Change the **backgroundColor** value to your desired color (e.g., **'# c71630'**).

```
57  const styles = StyleSheet.create({
58    titleContainer: {
59      flexDirection: 'row',
60      alignItems: 'center',
61      gap: 8,
62    },
63    stepContainer: {
64      gap: 8,
65      marginBottom: 8,
66      backgroundColor: '#c71630'
67    },
68    reactLogo: {
69      height: 178,
70      width: 290,
71      bottom: 0,
72      left: 0,
73      position: 'absolute',
74    },
75  });
```

4) Run the command "**npx expo start --tunnel"** to start the development server.
If the server is running, you can see the update immediately.



Steps to verify the changes:

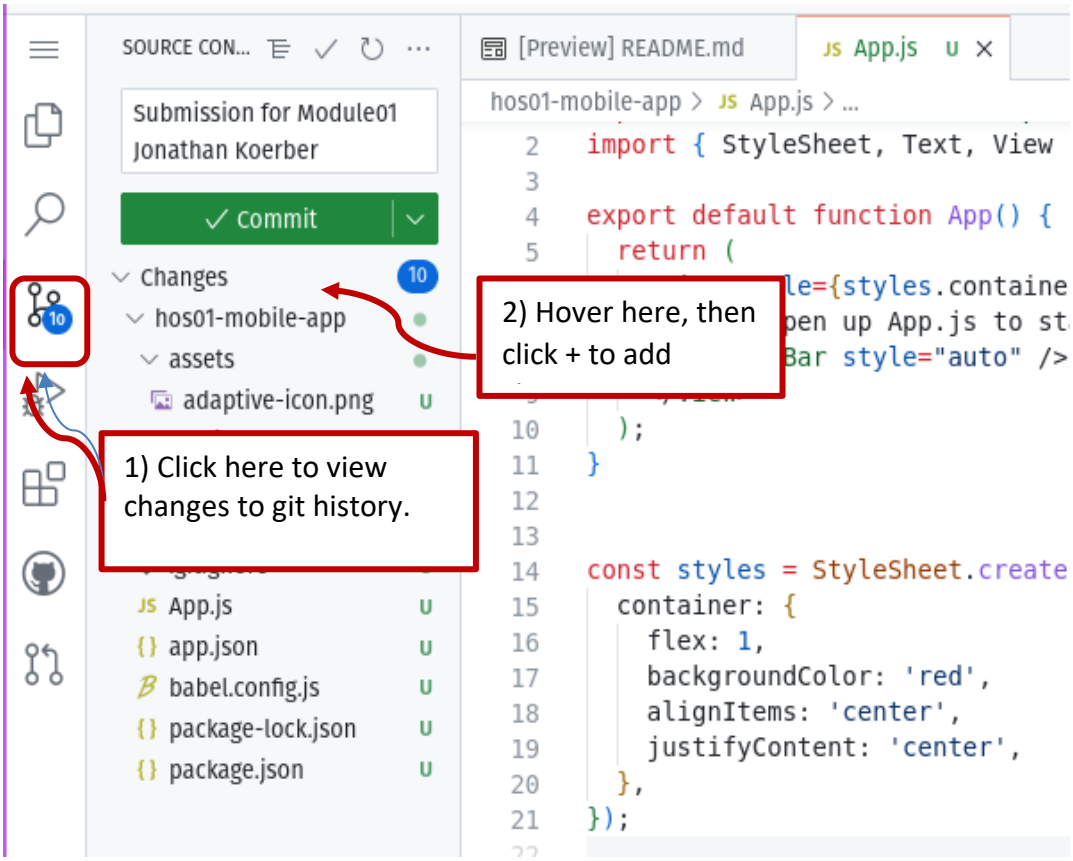
1. Open the terminal in the GitHub Codespaces environment.
2. Type **expo start --tunnel** and press Enter to start the expo development server.
3. Wait for the development server to load and show the QR code.
4. Open the "Expo Go" app from your mobile device.
5. Scan the QR code shown in the terminal with the "Expo Go" app or the Camera app (iOS).
6. Wait for the app to load on the mobile device.
7. Verify that the background color of the app has been changed to the desired color.

NOTE: Loading the app for the first time may take some time, and you may need to click Reload JS if you receive an error due to slow connections. In fact, if you use this method, the Codespaces and your phone may be disconnected due to inactivity, so if you see Disconnected from the Metro server, shake your phone to bring up the Expo devtool menu and click Reload.

5) What is the purpose of this section? i.e., why did you practice this section

Section 5: Pushing your work to GitHub

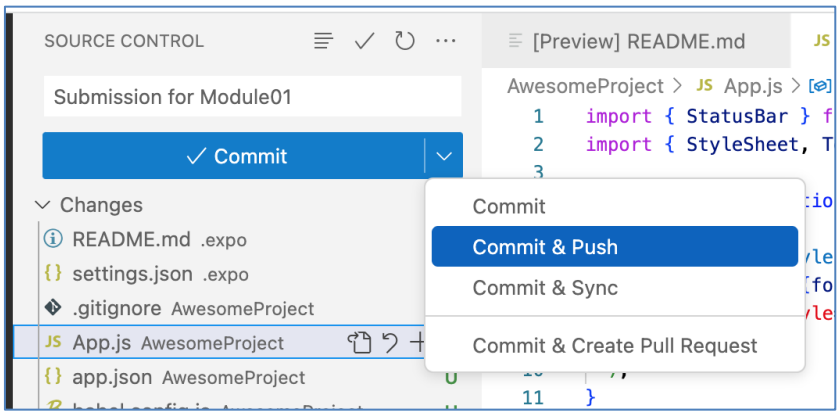
1) Go to Source Control on your GitHub codespace and observe the pending changes.



2) Type the Message for your changes in the Message box on the top. For example," **Submission for Module01 – Your Name**"

3) Click on the dropdown beside the commit button and select **Commit & Push** to update the changes to your repository main branch.

4) Select **Yes** when prompted.



5) What is the purpose of this section? i.e., why did you practice this section?