Your First App - Follow this Step By Step

Links:

1. Preface
2. Setting up your Environment
   1. Github
      1. Create an account
         1. <https://github.com/join>
      2. Create a repository A screenshot of a cell phone

         Description automatically generated
         1. Follow steps in the new repo. You can make it private if you want no one to access it, however, many, if not most, repositories are Public.A screenshot of a cell phone

            Description automatically generated
            1. Do not initialize a README. We will do this later via Terminal.
   2. Now, go to Xcode
      1. Create project following screenshotsA screenshot of a cell phone

         Description automatically generated
   3. A screenshot of a computer

      Description automatically generatedA screenshot of a social media post

      Description automatically generatedA screenshot of a social media post

      Description automatically generated
      1. Be sure to note what file location you saved it in. You need to be able to locate it via “Terminal” or “the command line” in the next step
      2. Now you need to set up your Github in Xcode. (Xcode calls it source control sometimes). First, click on **Preferences** in the dropdown in the upper left of your screen.

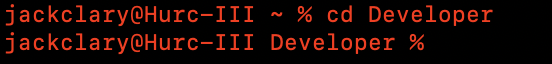
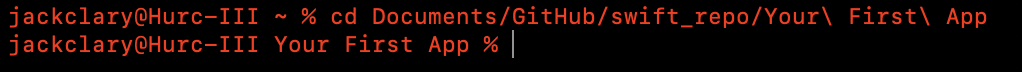
A screenshot of a cell phone

Description automatically generated

* + 1. Now, click “Source Control” in the upper center and make sure you have these boxes checked.A screenshot of a cell phone

       Description automatically generated
    2. Lastly click the little “Git” and type in your information
    3. A screenshot of a cell phone

       Description automatically generated
    4. You will have to type your Github account information later, but it can/will happen a few different times so just have it at the ready.
  1. Now, open Terminal
     1. “**ls**” shows current directory location and its contentsA close up of a logo

        Description automatically generated
     2. “**cd** ‘**filename’**” moves you to said file or directory
        1. As you can see, my “directory” changed from “~” to “Developer”
        2. If you type in “**cd** **..**” it will bring you back to the previous directory.
        3. If you type in “**cd**” it will bring you back to the “~” directory
     3. So, cd to the directory you have your Xcode project saved in
        1. If the title of the directory is unique, you can press the **tab** **key** after you’ve typed in sufficient letters to auto-type the rest. This is useful when there are spaces like in “Your First …” because they require “\ /” in terminal.
     4. Use the following commands in terminal when you are in your directory. You know you are in your directory when it says the correct name to the left of the “%” (E.g. “Your First App” in my case)A screenshot of a cell phone

        Description automatically generated
        1. The link, i.e. <https://github.com/jaclary/test.git> will be different for you based on your github username and the name of the repository. This link can be found if you look at your browser’s search bar when you are on your repository page which will load after you create it in step 2.a.ii.1

1. Your First App - Users can sign-up for your service
   1. After you start the project + initializing your repository
      1. Creating a .gitignore
         1. **“cd”** to the repository location in terminal
         2. “**touch** **.gitignore**”
         3. “**open** **.gitignore**”
         4. This is basically a text file where each line represents a file or files you don’t want being put in your repository.
      2. What to put in it?
         1. \*.xcuserstate
         2. \*.xcuserdata
         3. \*.DS\_Store
         4. The “\*” at the beginning means all files that contain the specified string.
      3. How to github?
         1. Terminal vs. xcode
         2. Xcode “Source Control”
            1. Save

Commit

Push

* + - * 1. Pull/Update

Pull

* 1. What are these initial files?
     1. AppDelegate
     2. SceneDelegate
     3. ContentView
     4. Assets.xcassets
     5. LaunchScreen.storyboard
     6. Info.plist
  2. SignUpView
     1. What are views in SwiftUI?
        1. VStack
        2. HStack
        3. Group
        4. Text
        5. TextField
     2. @State
        1. Storing user data
        2. Create a user class? (maybe throw all signed up users in a list that displays after they sign
     3. NavigationView?
  3. SignUpViewModel