Generic Stack View to work with ViewControllers
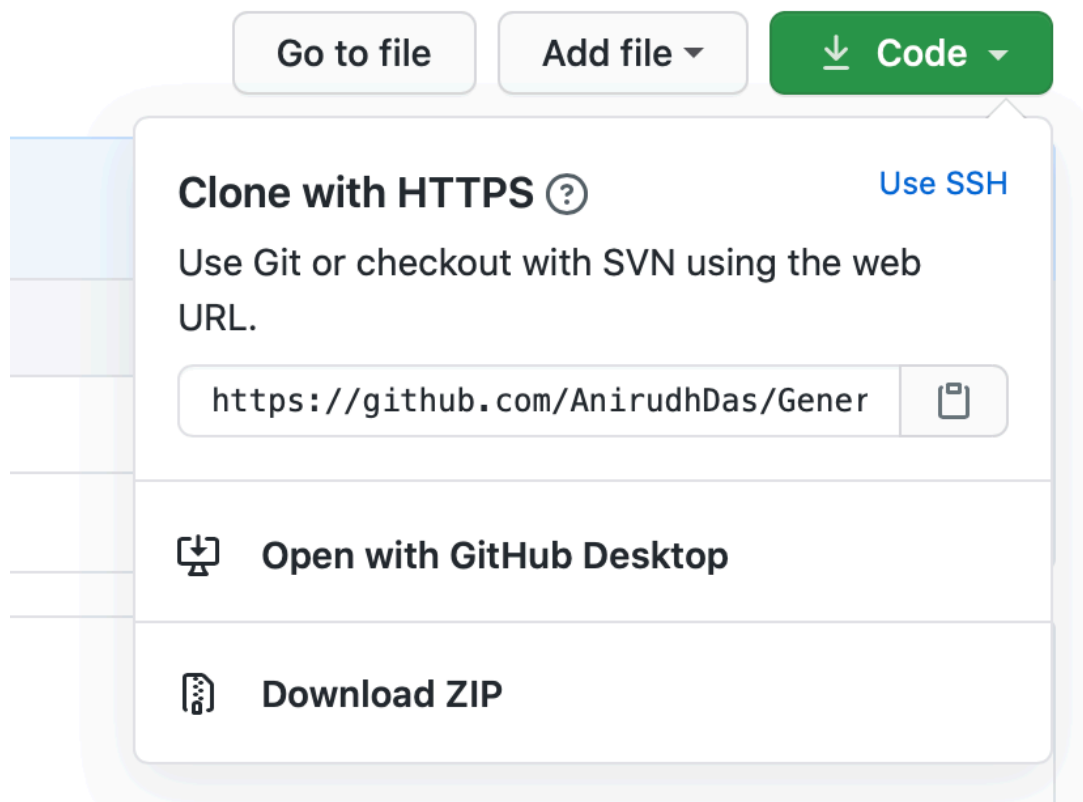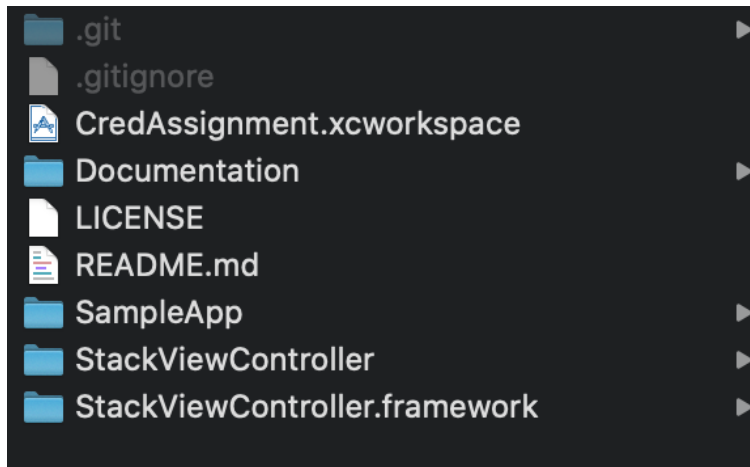
Where to download / checkout the source code?
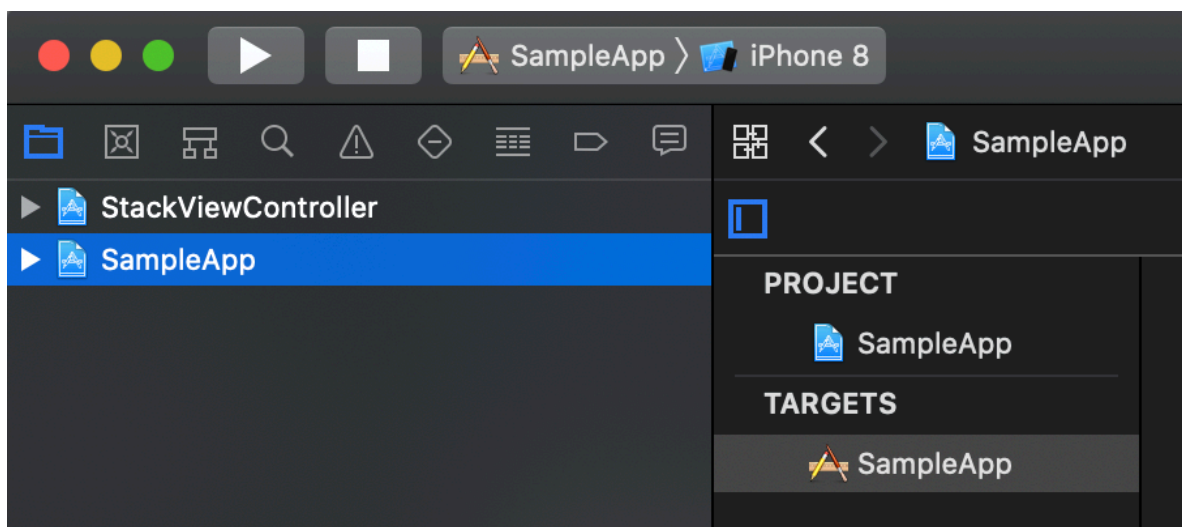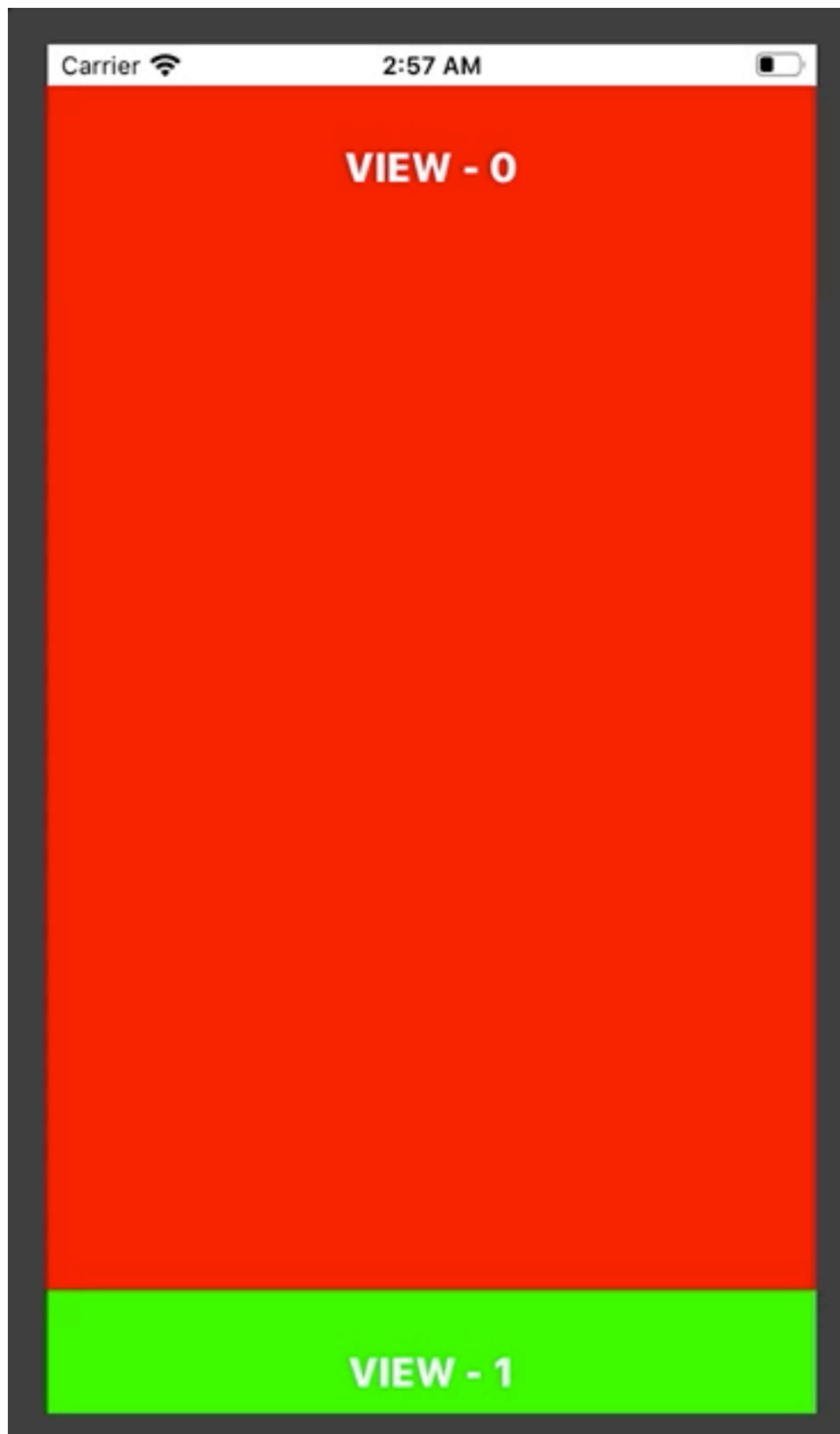**https://github.com/AnirudhDas/GenericStackView**

How to run the app?

1. Open *CredAssignment.xcworkspace*

2. This contains both the framework project as well as the sample app. Make sure to select the SampleApp target before running.

3. There you go! You can see the generic stack view in action.
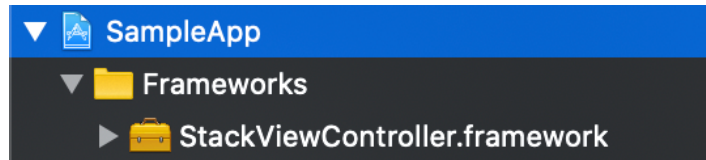
Development Tools used:
1. Xcode 11.3.1
2. macOS Catalina 10.15.4
3. Swift 5

Little bit about the approach taken:
I have created the generic stack view as a **<u>Dynamic Framework</u>** here. So that we can use it as a 3rd party dependency in other applications.

How to use it in other applications?
   1. In the sample app, drag and drop the *StackViewController.framework*

   2. Link the framework in the *Target -> General -> Frameworks, Libraries and Embedded content.*
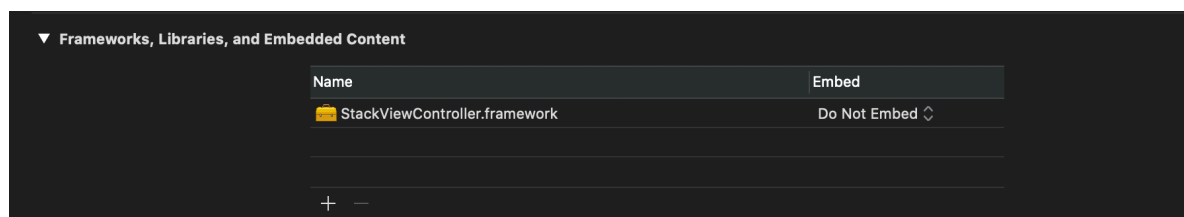
   3. To test the framework, in the AppDelegate file.
       i. *import StackViewController*
       ii. Create the array of ViewControllers and initialise the *StackViewController* with the array. You can pass any number of VCs to the initialiser.
       iii. Set the StackViewController as the *rootViewController* of the application window.

       For demo, i have created a ViewController called *TestVC.swift*, which looks something like this. You can use any ViewController of your choice.

HEADER
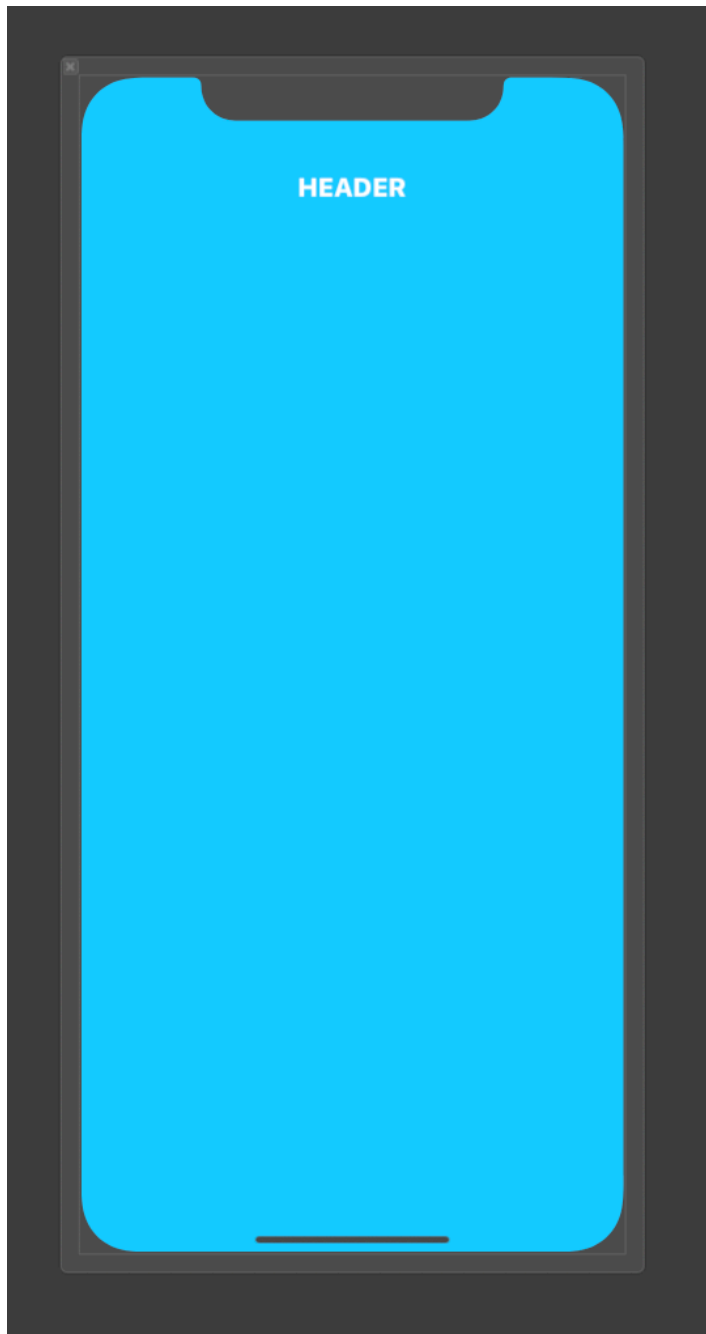
4. The app delegate looks like this:

```
import UIKit
import StackViewController

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    var window: UIWindow?

    func initialVC() -> UIViewController {
        //Add any custom VC.
        let vc1 = TestVC(position: 0, bgColor: .red)
        let vc2 = TestVC(position: 1, bgColor: .green)
        let vc3 = TestVC(position: 2, bgColor: .blue)
        let vc4 = TestVC(position: 3, bgColor: .gray)
        let vc5 = TestVC(position: 4, bgColor: .purple)
        let vc6 = TestVC(position: 5, bgColor: .brown)

        return StackViewController(subViewControllers: [vc1, vc2, vc3, vc4, vc5, vc6])
    }

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        self.window = UIWindow(frame: UIScreen.main.bounds)
        self.window?.rootViewController = initialVC()
        self.window?.makeKeyAndVisible()

        return true
    }
}
```

NOTE: The VCs can exist separately outside the stackview. We can use them independently.

Behaviour of the component:

1. By default, the view1 is expanded. And view2 is collapsed and peeks from bottom.
2. On click of view2, view 1 collapses and shrinks to top. view2 expands, and view3 is collapsed and peeks from bottom.
3. The same behaviour follows for remaining views.
4. Now, we can click and collapsed view, and it expands, causing the very next view to collapse and peek from bottom. While the remaining views are hidden. And the previous views are still collapsed.
5. All stack views components should have two states (Expanded and collapsed). Clicking on any collapsed view toggles its state i.e. it expands and the other expanded view, collapses.