

Home assignment #2

Due date: 20.07.2022, 23:59

Home assignment consists of 3 separate assignments:

1. Styling launch (splash) screen,
2. Adding app icon and
3. Layouting login screen.

You can find all screens and design on [this link](#). It will open Figma tool which is used on production apps. There you can inspect UI elements, their distance and size, frames... You'll get all images and icons exported on materials repository, but for example on production app you would export it from Figma.

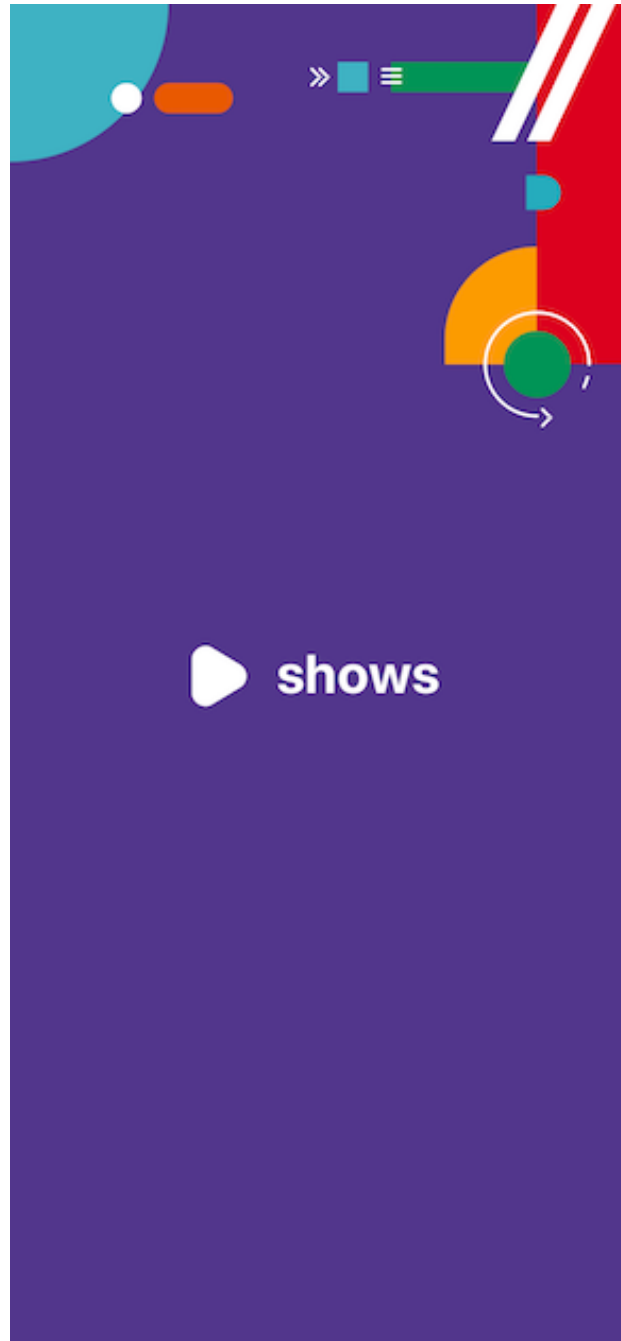
You'll probably use Figma to get the colors of backgrounds, labels, buttons and stuff. But for example, purple color below is

#52368C

or in RGB R: 82 G: 54 B: 140

1. Assignment - Styling launch (Splash) screen

This is how your splash screen should look like:

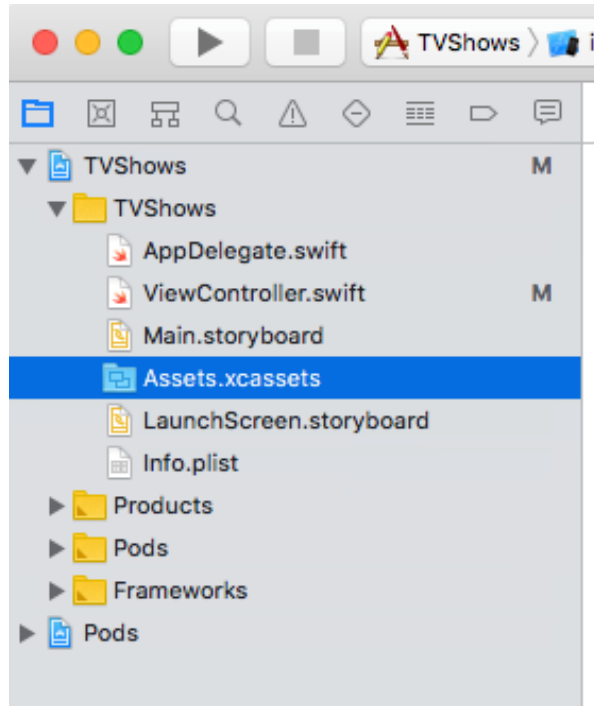


You will find the splash logo and top images inside the `design/images` folder on repository. Use `UIImageView` and center it vertically and horizontally using auto layout for logo, and top images via top and leading/trailing constraints

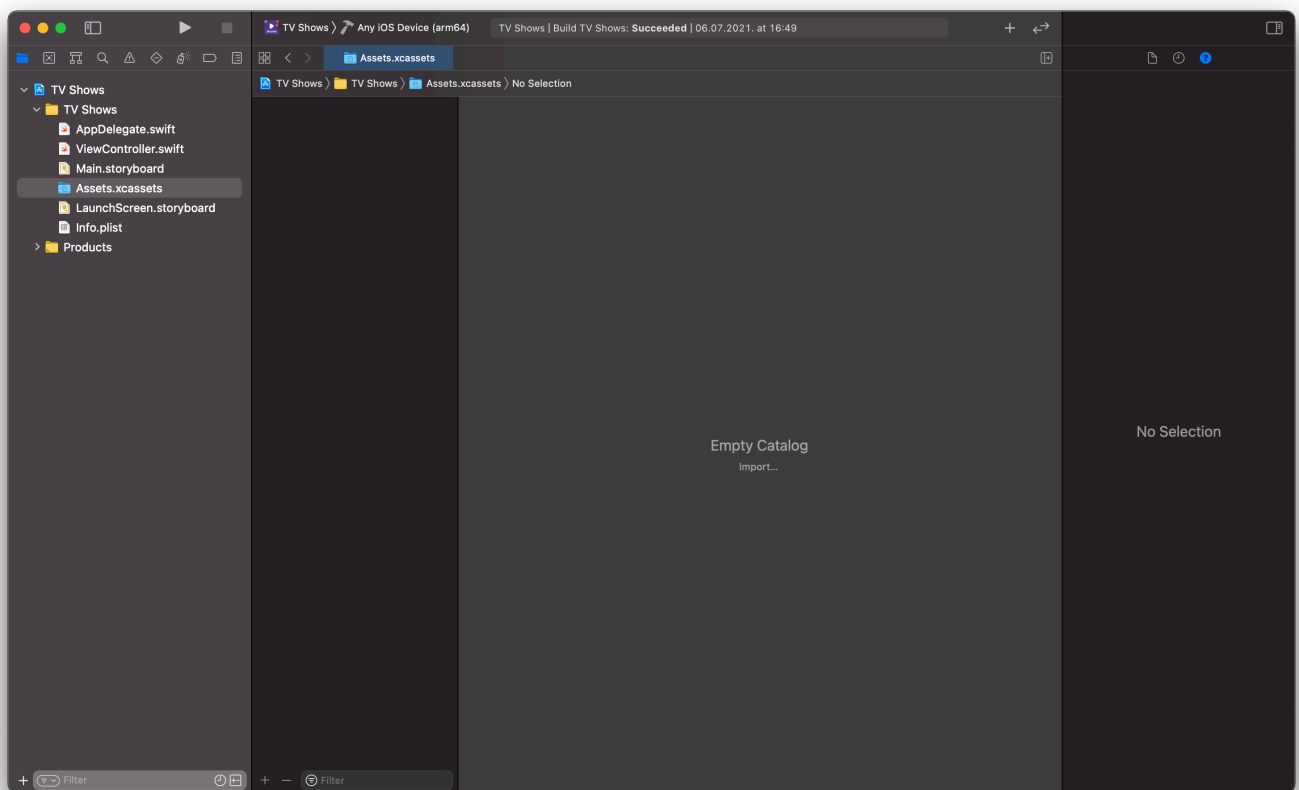
But before that, here is a short tutorial on how to add an asset.

Adding assets

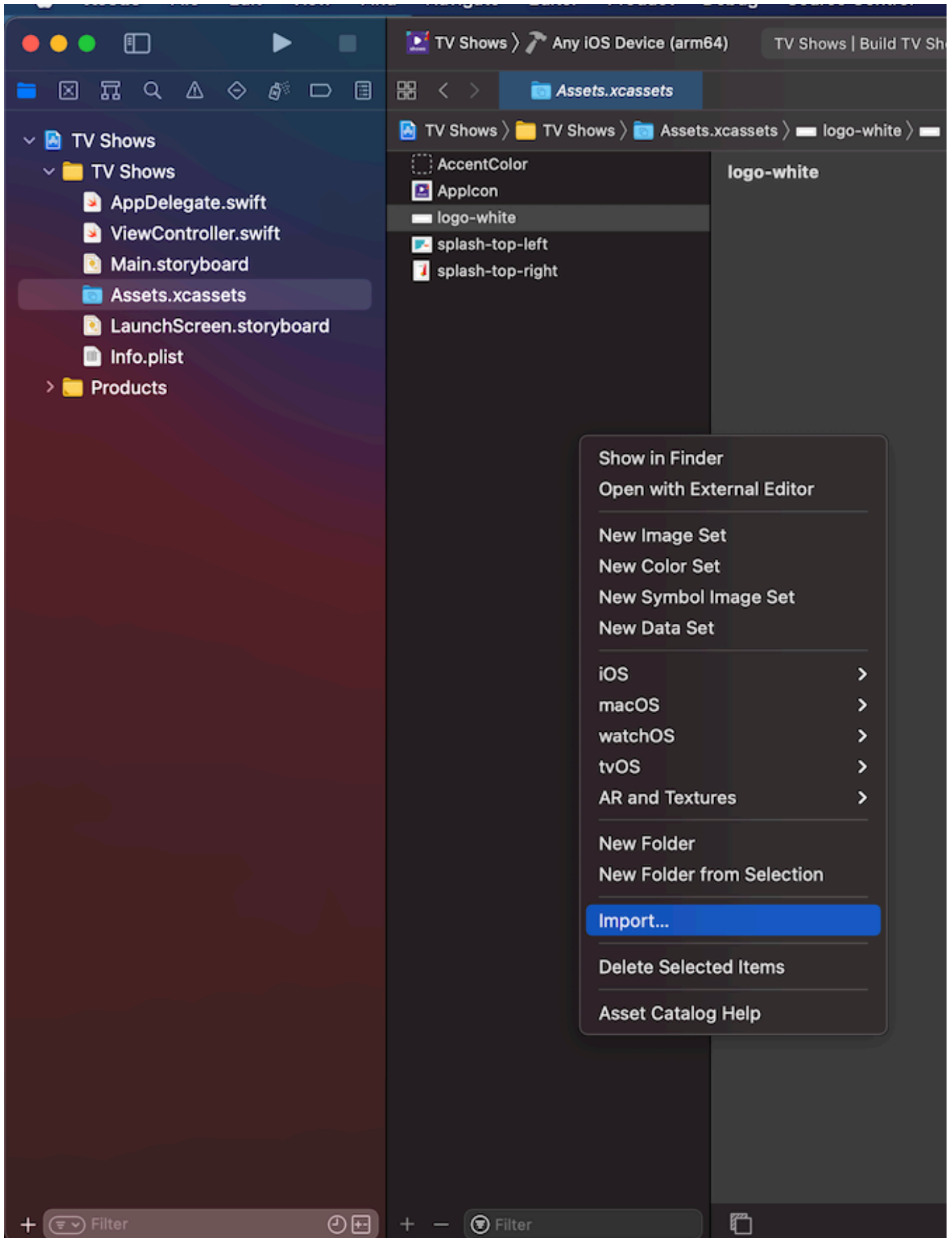
Special folder structure called **.xcassets** is used for importing and storing assets (images, app icon) in iOS app. When you created your Xcode project it automatically generated **.xcassets** folder for you. You should see it inside Xcode in the root folder of your project.



Select **Assets.xcassets** and you should see something like this:

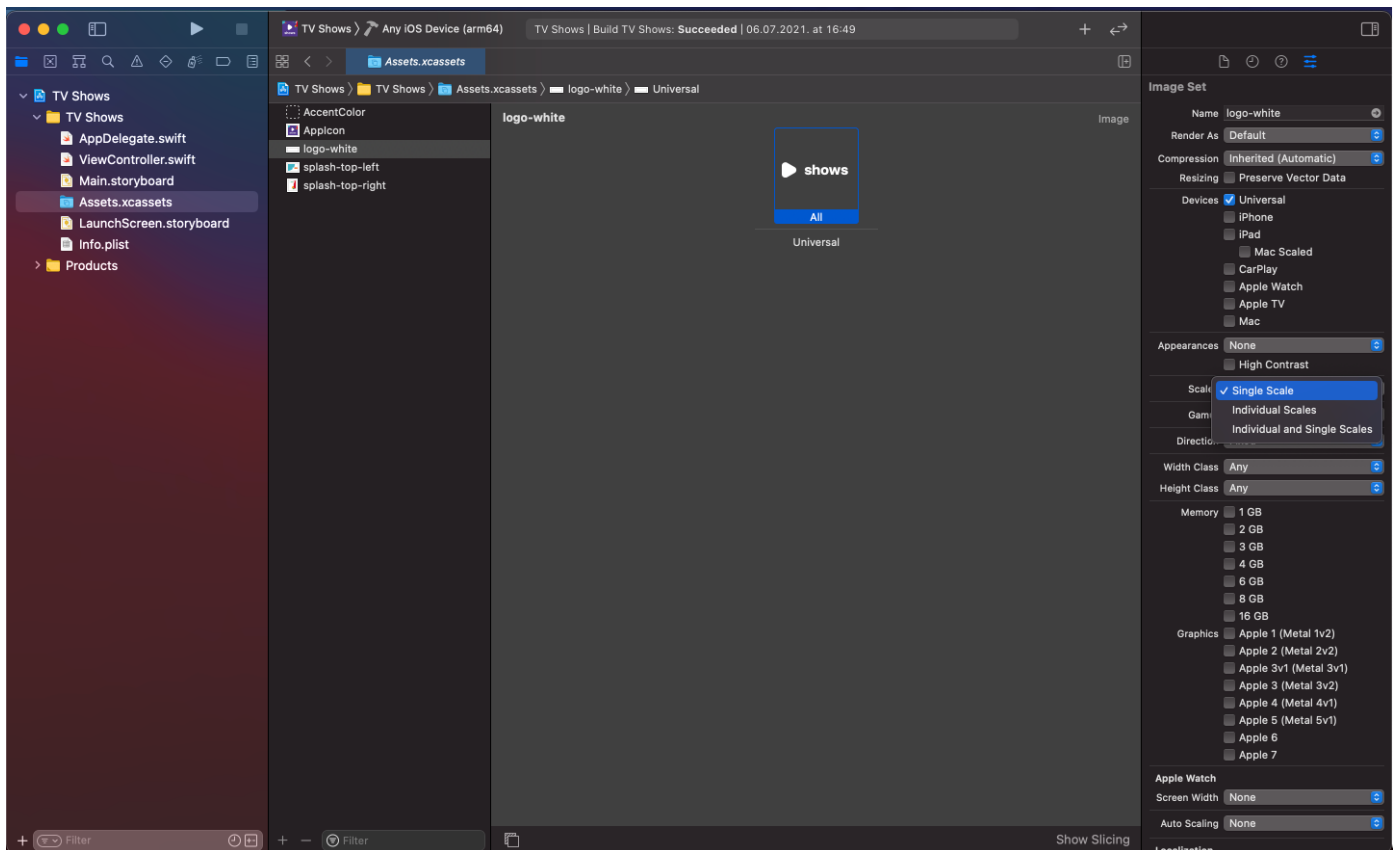


Right click on an empty part of the section where is *AppIcon* should give you dropdown menu with **Import** command:



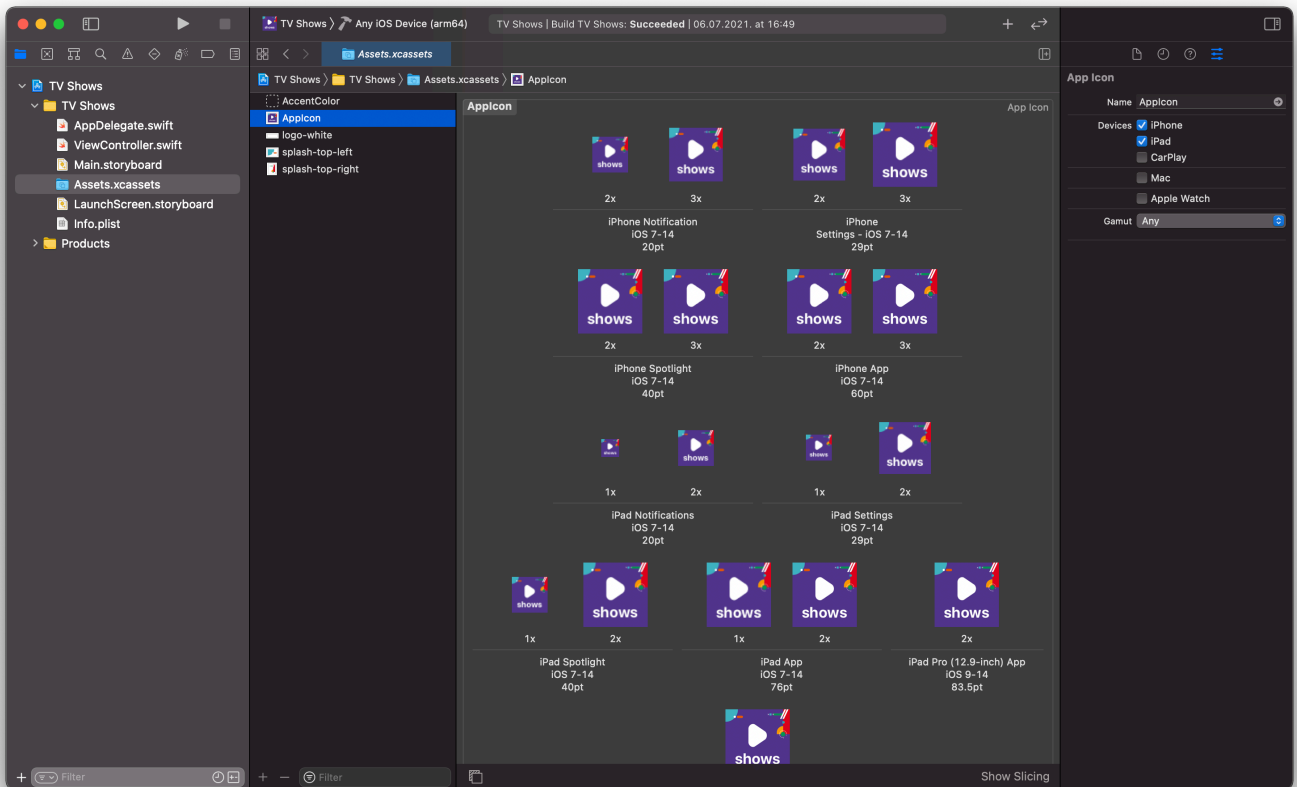
Pressing on `Import...` you'll get file explorer (Finder) where you should select your PDF or SVG file or multiple select 1x, 2x, 3x image sizes.

Also, you can drag and drop them in the same section where is *AppIcon* but Xcode has issues with vector graphics imported in that way (you'll need to set `scales` property for each vector image to `Single scale` - select image -> go to Utility pane (right part of screen)).

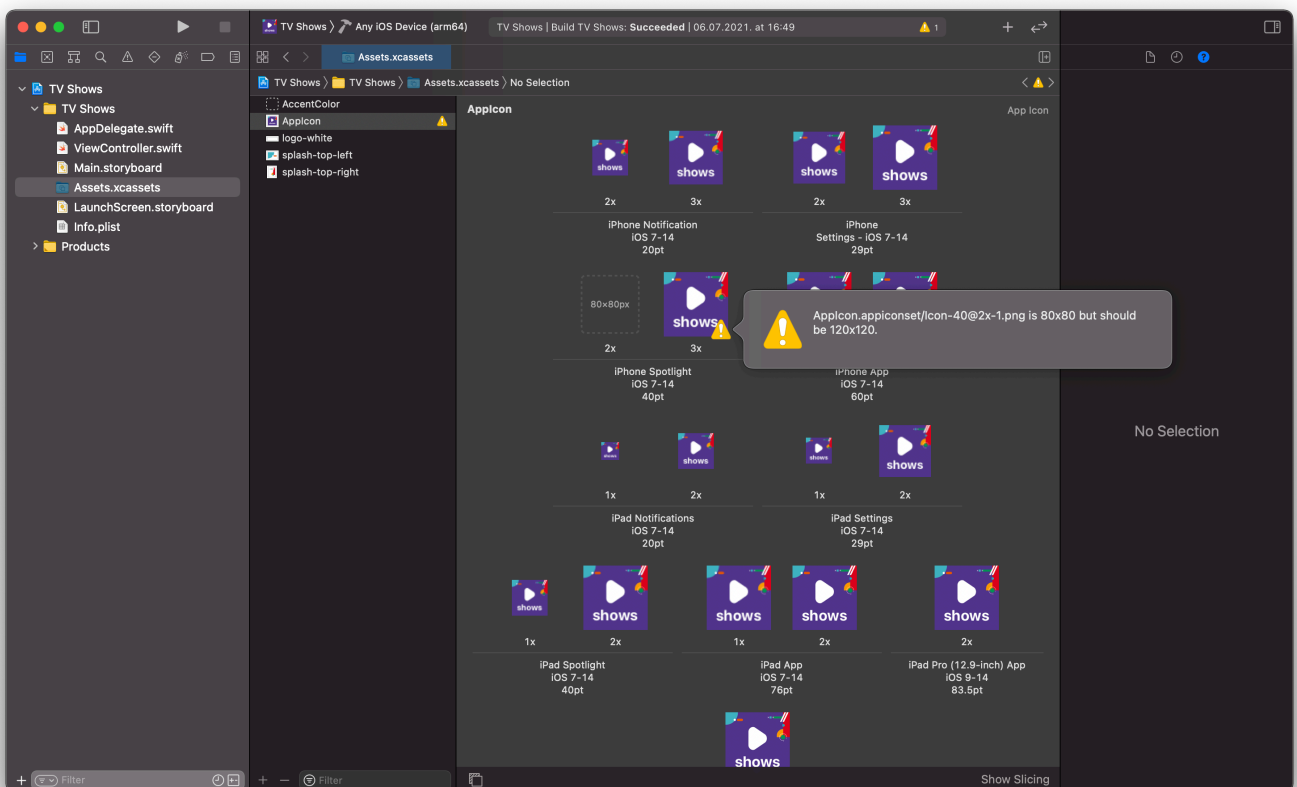


2. Assignment - Adding an app icon

An app icon is also added through the assets folder using *AppIcon* asset. When you open it for the first time you should see empty *AppIcon* asset. There you should put app icons given in **app-icon** folder. Putting icons is done easily with drag 'n' drop from folder to the correct box. I gave you rightly named icons (base resolution @ scale) so it should not be a problem to match icon name with box name. What matters is that you need to match icon size with the box where you putting it (every iPhone and iPad have different resolutions, because of that @2x and @3x suffixes are used - to distinguish for which resolution is used). If everything goes okay you should get a screen something like this:

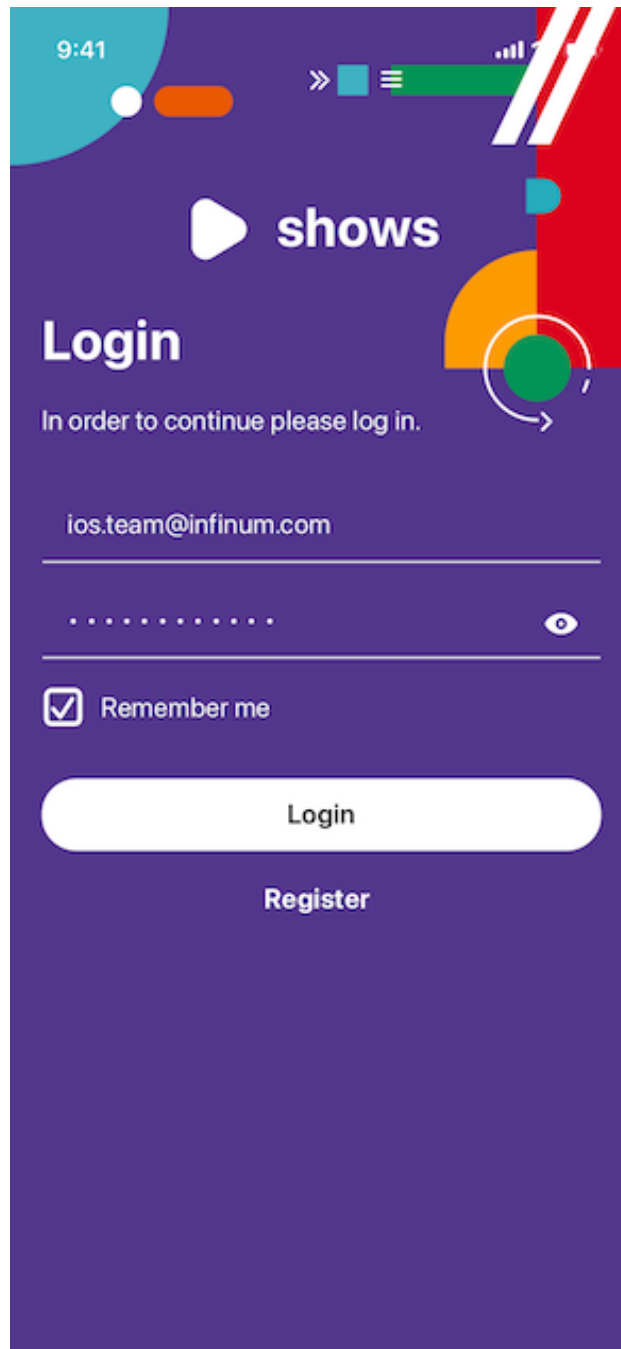


Now you run your app. If you put icons in the right place you should see TV Shows icon in Simulator as your app icon. And most important, check if Xcode produced some warnings about given images in Project status. If it isn't, then you successfully added app icons. If you made a mistake, Xcode will warn you:



3. Assignment - Layouting login screen using AutoLayout

This is how your login screen should look like:



As help here is the list (in y-axis order) of UI elements which you **could** use:

- `UIImageView` - logo, top images
- `UITextField` - username input (hint: placeholder)
- `UITextField` - password input (hint: placeholder)
- `UIButton` - checkmark (hint: use different icons for each state - checked and unchecked);
- `UILabel` - Remember me
- `UIButton` - background and corner radius (hint: radius should be set from code using `layer` on button)
- `UIButton` - without background

Two used colors are: #FFFFFF - white and #52368C - purple

Be creative with constraints :)

4. extra assignment

UIStackView

Try to create Login screen layout using UIStackView.

Feel free to use multiple stackviews, stackviews inside stackviews (pretty common thing to do), everything is possible :)

UIScrollView

UIScrollView, as its name states, is a scrollable container. It is used to enable content scrolling when content size is bigger than the screen size. It supports vertical and horizontal scrolling.

Few tips when working with UIScrollView:

1. You should wrap all your components inside one `UIView` - e.g. same as you add UIButton to View Controller, you can add it to UIView - it is just empty placeholder for other views. It is easier to manipulate with constraints.
2. Scroll view needs to know its content size (width/height). You can constraint your container view to have the same width as scroll view, and use top/bottom constraints among inside components to tell UIView how much height does it need.
3. Step to add UIScrollView - check out our example with scroll view - take look at UI elements and their constraints.

Check out our `UIScrollView` example project and try to replicate it in your Login screen (for landscape mode when content is too big to display)