

 **IvanCaldwell / ios-sprint1-challenge**
forked from [LambdaSchool/ios-sprint1-challenge](#)

Sprint challenge for Sprint 1 of Lambda's iOS course

Edit


[Manage topics](#)

4 commits1 branch0 releases3 contributors


Branch: masterNew pull requestCreate new fileUpload filesFind fileClone or download


This branch is even with LambdaSchool:master.


Pull requestCompare


 armadsen and erica Merge pull request [LambdaSchool#30](#) from WilliamBundy/master ...


Latest commit a232cca on Oct 5

 [Movie List.xcodeproj](#) Merge pull request [LambdaSchool#30](#) from WilliamBundy/master 7 days ago

 [Movie List](#) Merge pull request [LambdaSchool#30](#) from WilliamBundy/master 7 days ago

 [.gitignore](#) Add .gitignore 4 months ago

 [README.md](#) Merge pull request [LambdaSchool#30](#) from WilliamBundy/master 7 days ago

 [Screen Recording.mp4](#) Add starter project and README 4 months ago

README.md

Sprint Challenge: iOS Fundamentals 1 - Movie List

This challenge allows you to practice the concepts and techniques learned over the past week and apply them in a concrete project. This Sprint explored iOS Basics. During this Sprint, you studied the Xcode IDE, Model-View-Controller, container controllers, and table views. In your challenge this week, you will demonstrate proficiency by creating an application that collects and lists your favorite movies.

Instructions

Read these instructions carefully. Understand exactly what is expected *before* starting this Sprint Challenge.

This is an individual assessment. All work must be your own. Your challenge score is a measure of your ability to work independently using the material covered through this sprint. You need to demonstrate proficiency in the concepts and objectives introduced and practiced in preceding days.

You are not allowed to collaborate during the Sprint Challenge. However, you are encouraged to follow the twenty-minute rule and seek support from your PM and Instructor in your cohort help channel on Slack. Your work reflects your proficiency in iOS and your command of the concepts and techniques in this first unit.

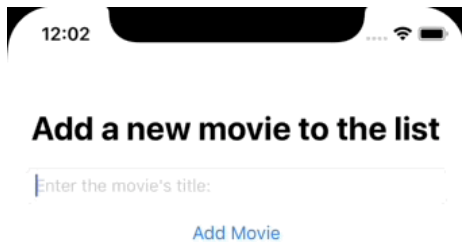
You have three hours to complete this challenge. Plan your time accordingly.

Commits

Commit your code regularly and meaningfully. This helps both you (in case you ever need to return to old code for any number of reasons) and your project manager.

Description

In this challenge, you build an app that tracks and remembers your favorite movies. Your finished project looks something like this:



In meeting the minimum viable product (MVP) specifications listed below, your application will be able to add movies to a central model and then present them in a table view in a separate tab.

Project Set Up

This repository contains a basic project that includes art for your tab bars and a stylish launch screen. Use this project as the basis for your work.

Minimum Viable Product

Your finished project must include all of the following requirements:

- A tab bar controller with two child controllers. The children are:
 - A view controller that allows the user to add new movies
 - A view controller that shows a list of movies that have been created. Just like you have done in class this week, this movie list view controller is implemented using a subclass of `UIViewController`, **not** `UITableViewController`.
- Users can swipe to delete movies

- Users can reorder movies. Use the same Edit/Done button approach as you did in class.
- The list of movies is preserved between subsequent runs of your application

In your solution, it is essential that you follow best practices and produce clean and professional results. Schedule time to review, refine, and assess your work and perform basic professional polishing including spell-checking and grammar-checking on your work. It is better to submit a challenge that meets MVP than one that attempts to much and does not.

Validate your work through testing and ensure that your code operates as designed.

Stretch Problems

After finishing your required elements, you can push your work further. These goals may or may not be things you have learned in this module but they build on the material you just studied. Time allowing, stretch your limits and see if you can deliver on the following optional goals:

- Stylize the "Add Movie" screen so it doesn't look quite so [Soviet Brutalist](#). Make sure your design still looks right on other platforms and orientations.
- Introduce a way to edit the name of a movie.
- Add a custom `UITableViewCell` subclass with a "Seen"/"Not Seen" button. Tapping the button toggles whether the user has seen the movie or not.

Progress Dashboard

-

Download checked items