

PROJECT #4

The Problem

This week's project we'll be working with creating a table view that displays information about your friends. Each cell in the table view should show the person's name and age. Take some time to think about how you want to model this data and then how you'll display it in the table view. Don't forget about the principles of MVC in your design. For testing purposes, make sure you put at least 6 people on your list. If you don't have that many friends, make some up. Hint: Every person should be represented as a single entity, with both name and age.

Carrier	3:52 AM	
Barney is 16 years old		
Andy is 20 years old		
Lucy is 25 years old		
Frank is 27 years old		
Marie is 28 years old		

Challenge Problems

A hint for all of these challenge problems. Check out the method names in both the **UITableViewDataSource** and **UITableViewDelegate** protocols, the nice thing about UIKit is that the method names read like English.

1. Just for fun, make every even row have a height of 40 and every odd row a height of 60.
2. Let's add an additional view controller so that we can select one of our friends in the table view and then edit his name and age values in the detail view controller.
3. Add an image of your friends to the cell. If you want you can simply manually add images into your project to reference different people, that's fine. Remember to reflect this change in your model if you are keeping track of images of people. You can also figure out how to take images from the camera roll, but this is quite a bit trickier and will involve some research.
4. Sometimes in life we meet new people that we can now call our friends. The project so far consists of static data, meaning that you hardcode in who your friends are and you can never change that (unless of course you recode your application). Let's add the ability to add new friends to this table. A common way to do this is to add a *bar button item* with the add identifier, but a normal button will do as well.

5. Sometimes in life we stop talking to old friends and it can get to the point where they don't belong on our awesome friends' list app. See if you can implement the deletion mechanism in table views.
6. The last challenge for you is to create sections in your table view. Add a list of enemies that you want to display in your table and put them in a separate section. Even better, put section titles on each section to indicate "Friend" and "Foe". Hint: this requires reworking some of the current method implementations. Go wild!

Reference

- Lesson #4
- UITableView class reference
- UITableViewDataSource and UITableViewDelegate protocol references
- The Swift Programming Language book