**How to Create a Tip Calculator in iOS using Storyboards?**

UI designing in iOS can be done either by using SwiftUI or UIKit. Both have their own pros and cons. In this tutorial you will learn how to design basic UI designs using storyboards and UIKit. The UIKit is usually used in conjunction with storyboards. The Storyboards are used to graphically design the app screen providing a visual representation.

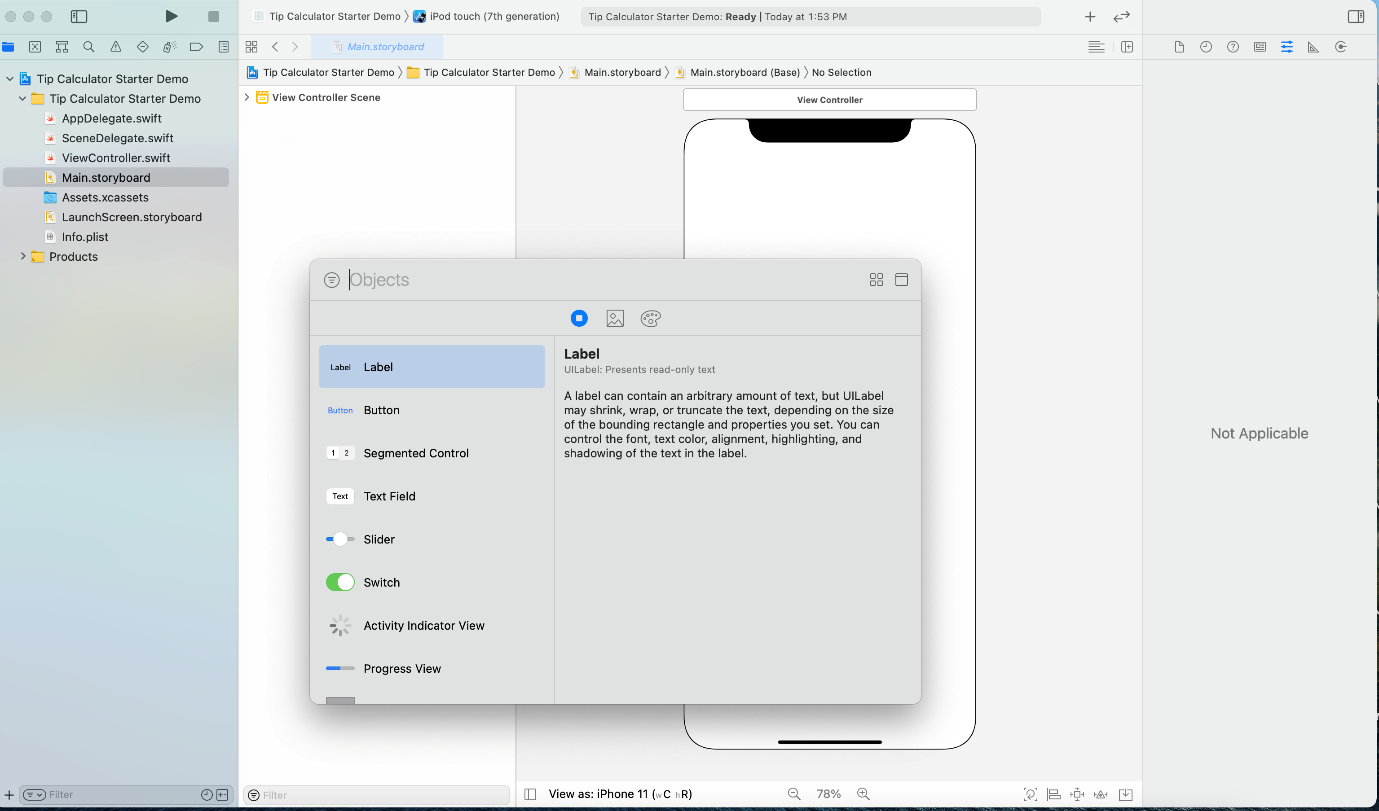
In this app, we are designing a tip calculator. Here the user enters the bill amount and the tip percent. The app calculates and returns the value of the tip amount and the total amount of the bill incurred.

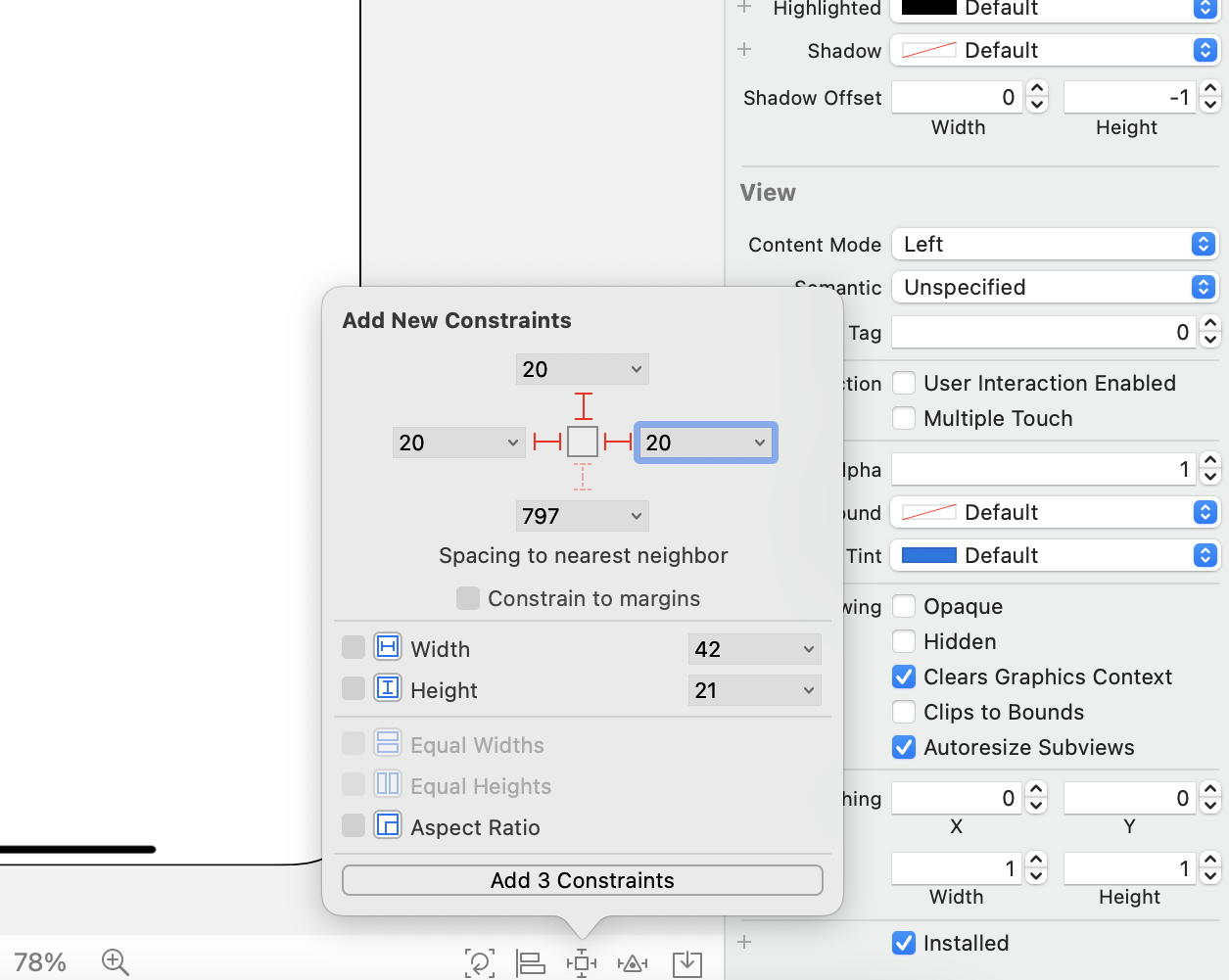
Getting Started

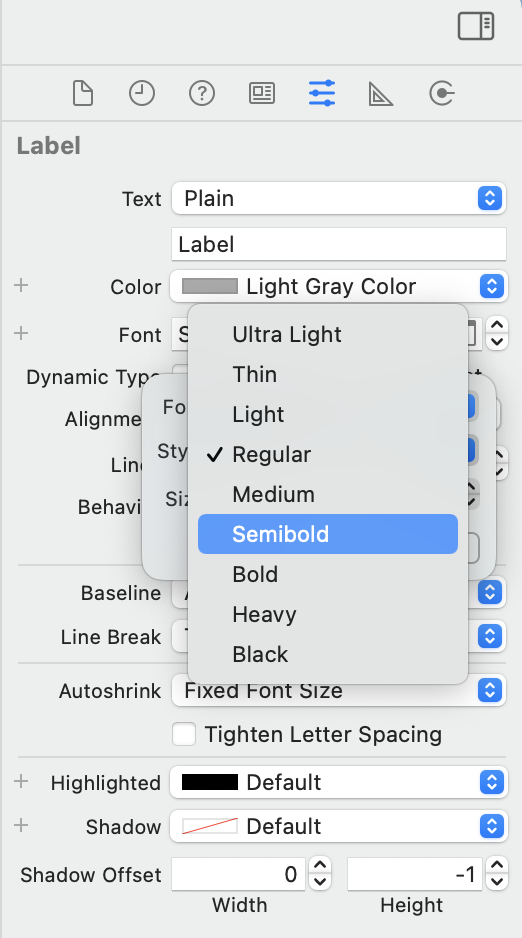
Download the starters project. Build and run it in Xcode. You will see the splash screen image set on LaunchScreen storyboard/

So Let’s Start

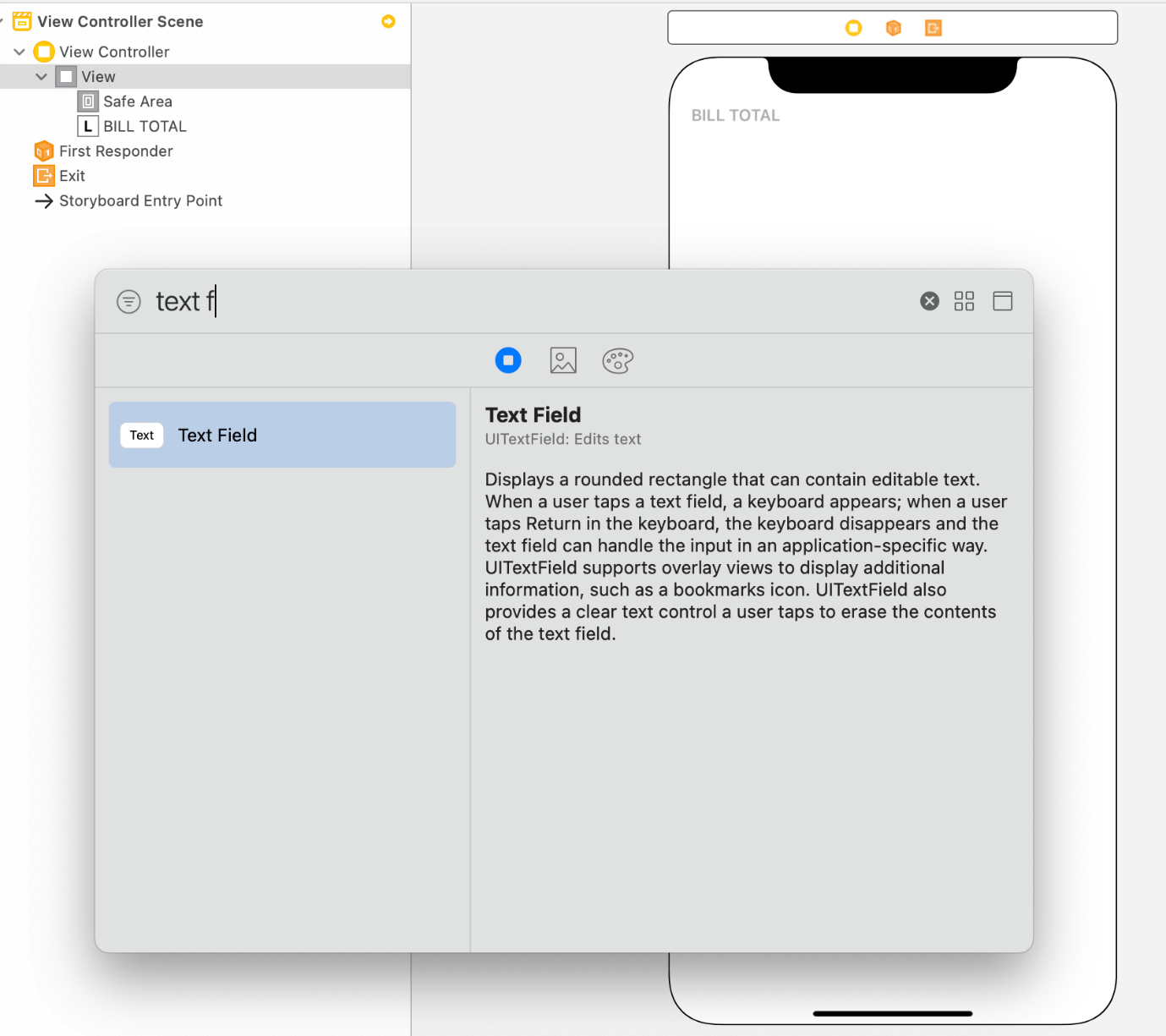
So, the first task is to design the UI. Go to the Main Storyboard. We are going to design the UI on the view controller depicted there.

Open the object library. Search for ‘label’. Drag and drop it to the top of the screen.Set its constraints the way shown below.

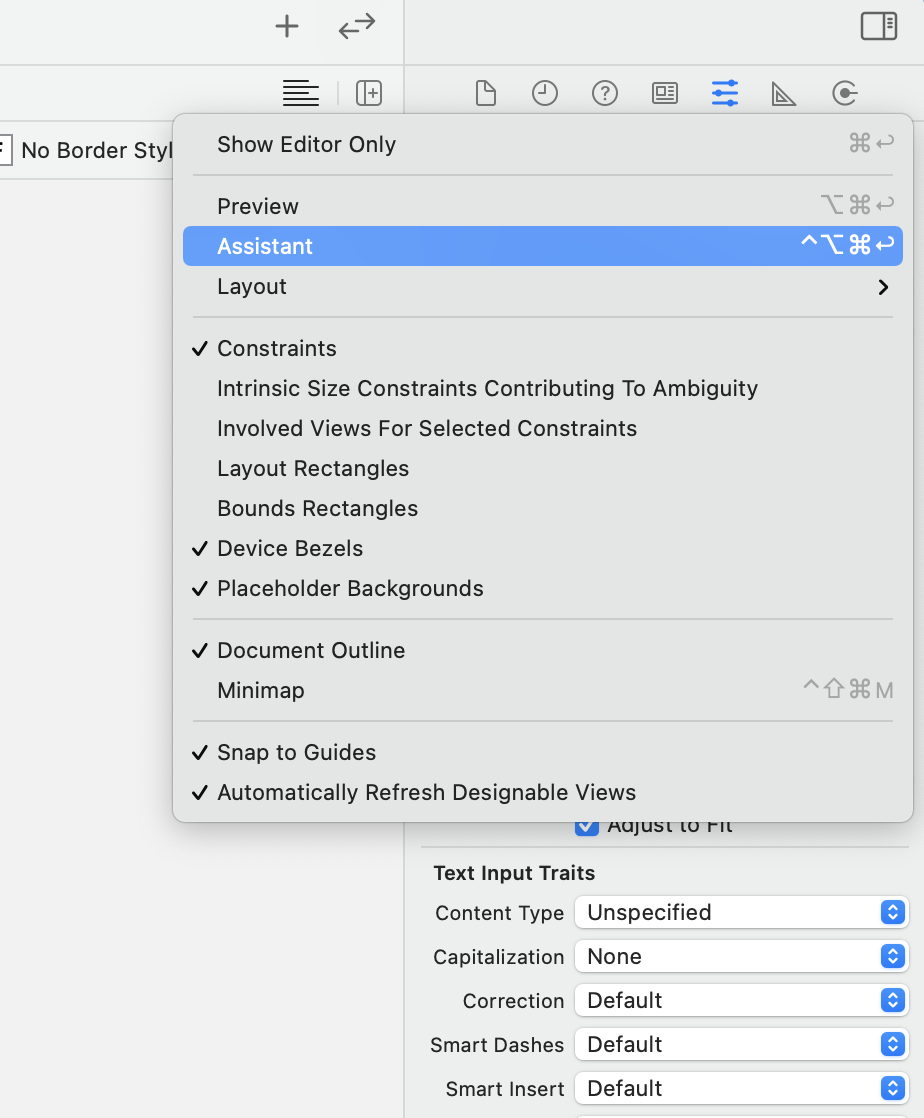
Now we need to change the attributes of the label. So go to the attribute inspector in the right side. Write ‘BILL TOTAL’ and change the following attributes as below.



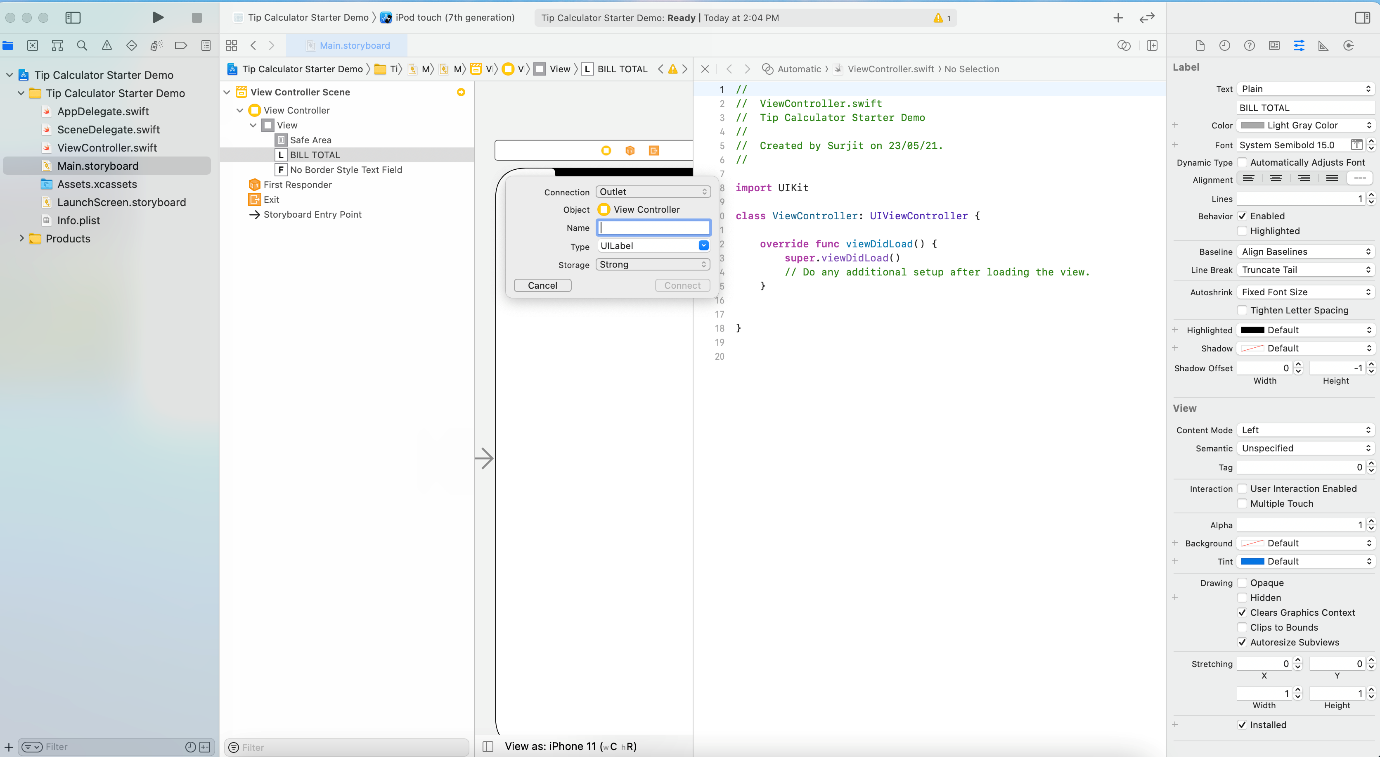
Similarly, open the object library and bring the textfield to the view controller just below the label. Put the constraints as before.

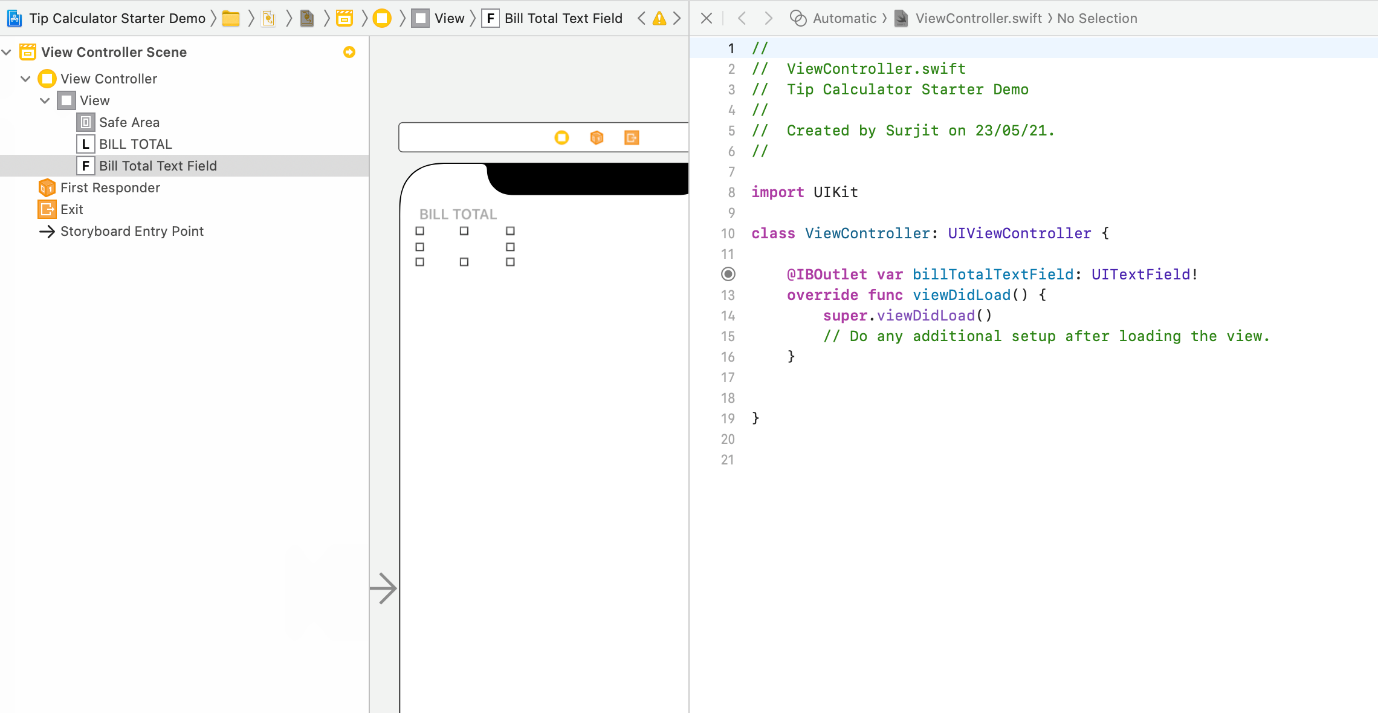


Go to the top right corner where a number of lines icon is showed. Click on it, select assistant from the options.



You will the code editor opens in your right side. Right click your label, drag and drop on the editor as shown below.





Repeat the above process to design the tip field.

Just below the label bring in button from object library. Place it below the SPLIT label, on extreme left. Give the following constraints –

Width = 48

Leading to safe area = 20

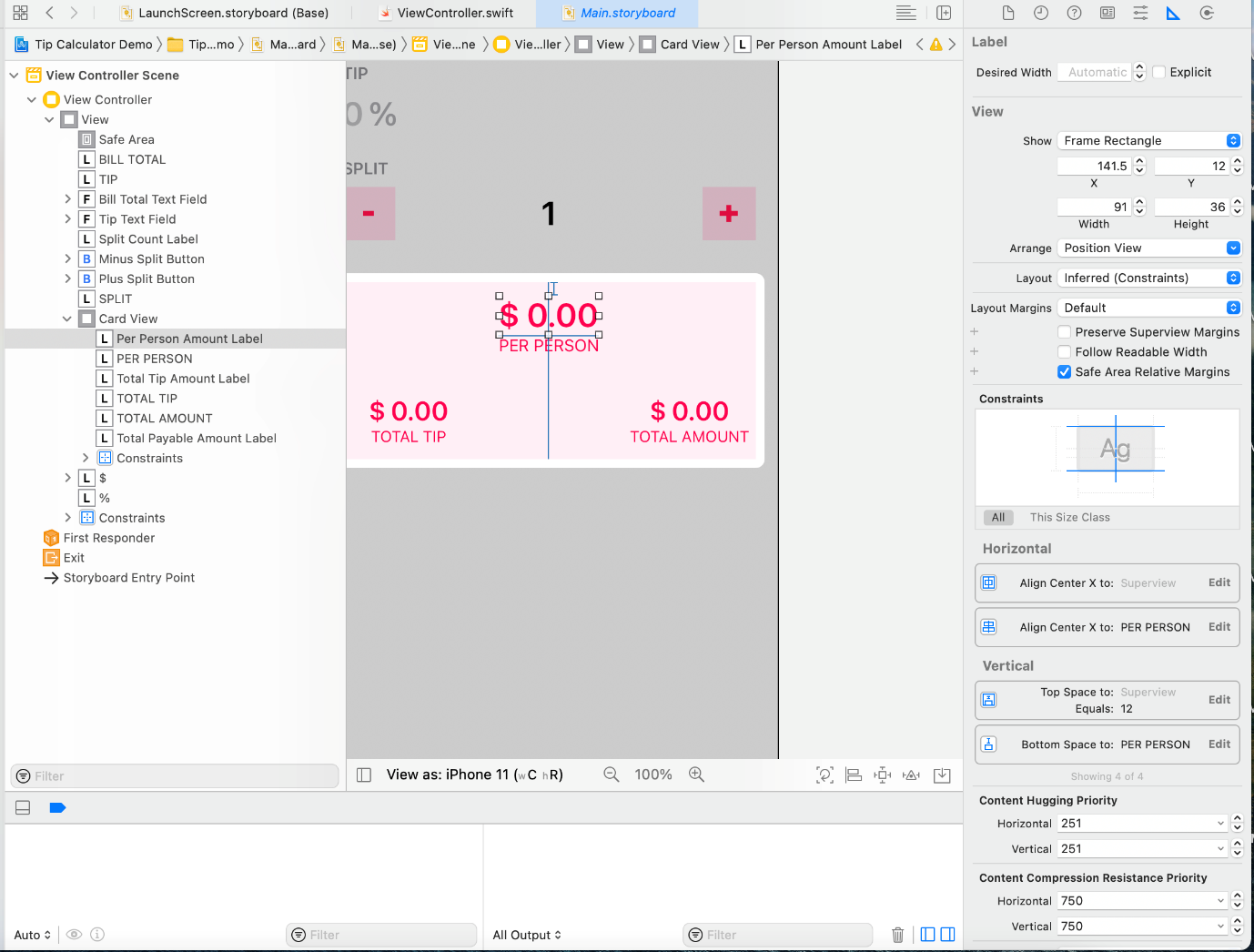
Height = 48

From the attributes mark the text color “system pink”. For background, go to custom -> RGB sliders -> put the hex color code as

Similarly bring another button giving similar width, height but only changing the trailing to leading constraints and keeping all its attributes constant. Right click, drag and drop from the right button to the left and select center vertically.

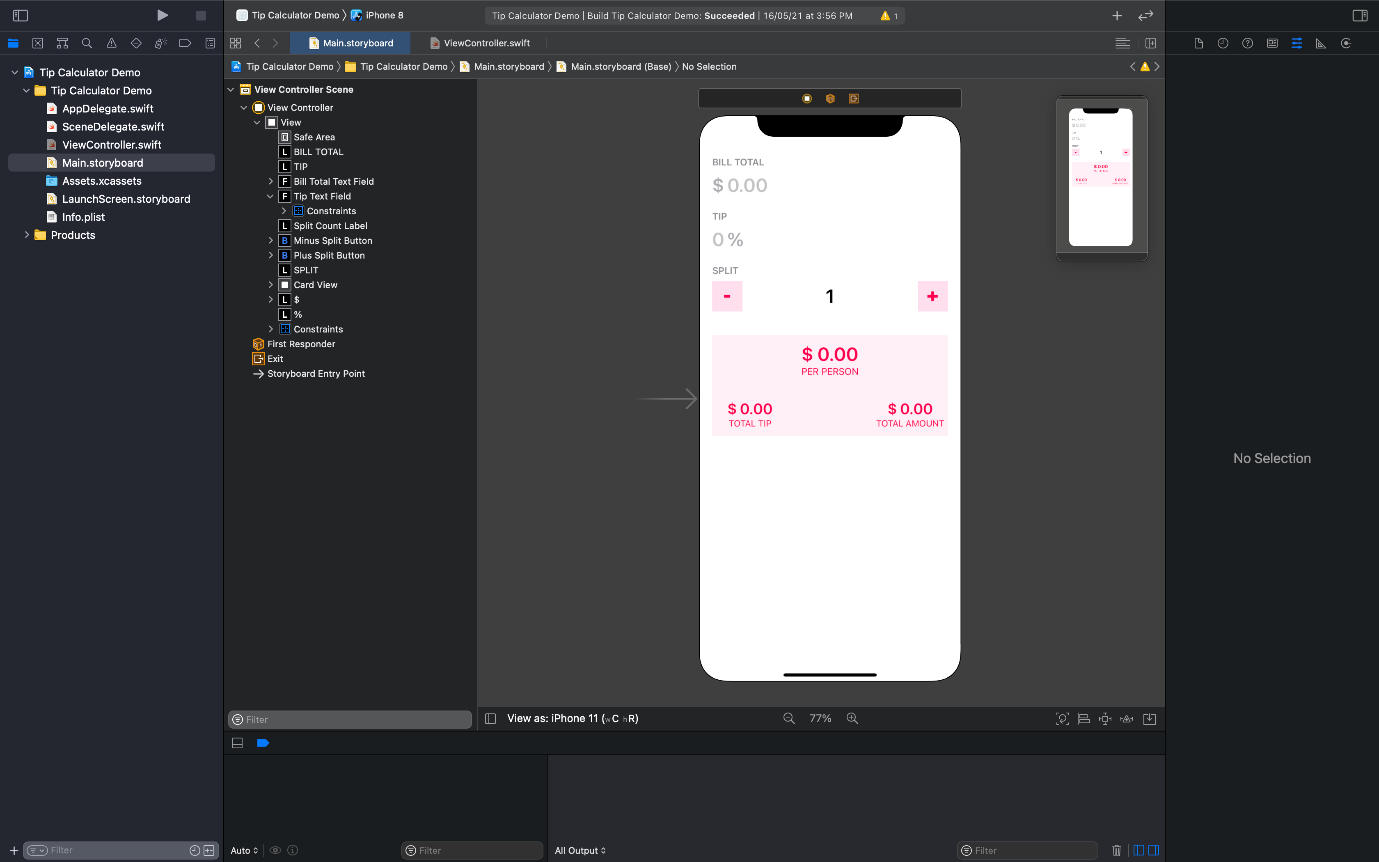
Next, drag drop another label and place it in between the 2 buttons. Follow the above step to put it centre vertically to one of the buttons.

Now we design the card view to display results. Bring UIView from object library, give top constraint 12 to the label above, height = 160 and both the leading and trailing constraints as 24. Inside the card view bring in the 6 labels and put the following constraints. Place 2 on the top of the card view, 2 on the left one above the other, and 2 on the right in similar pattern.

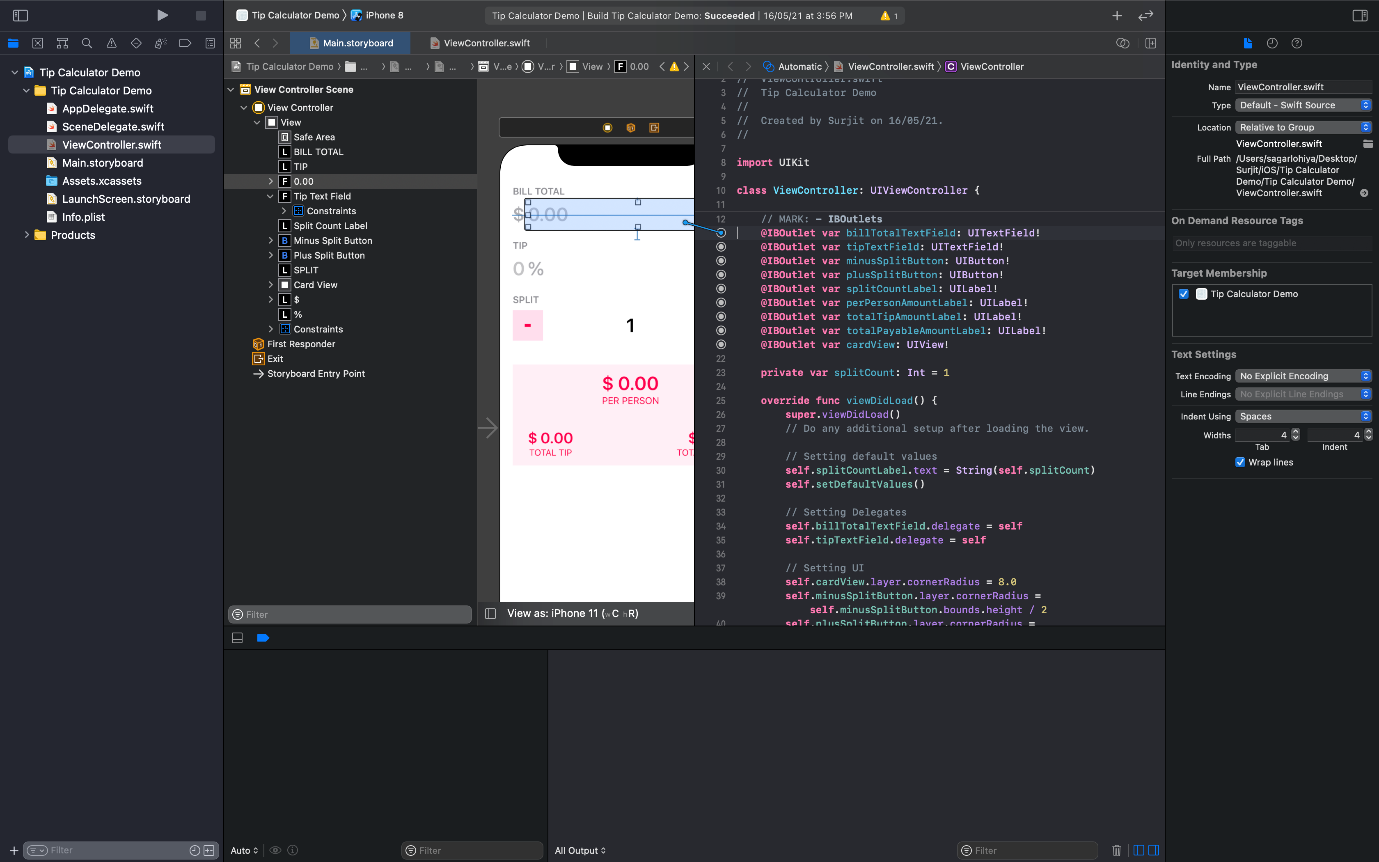


….

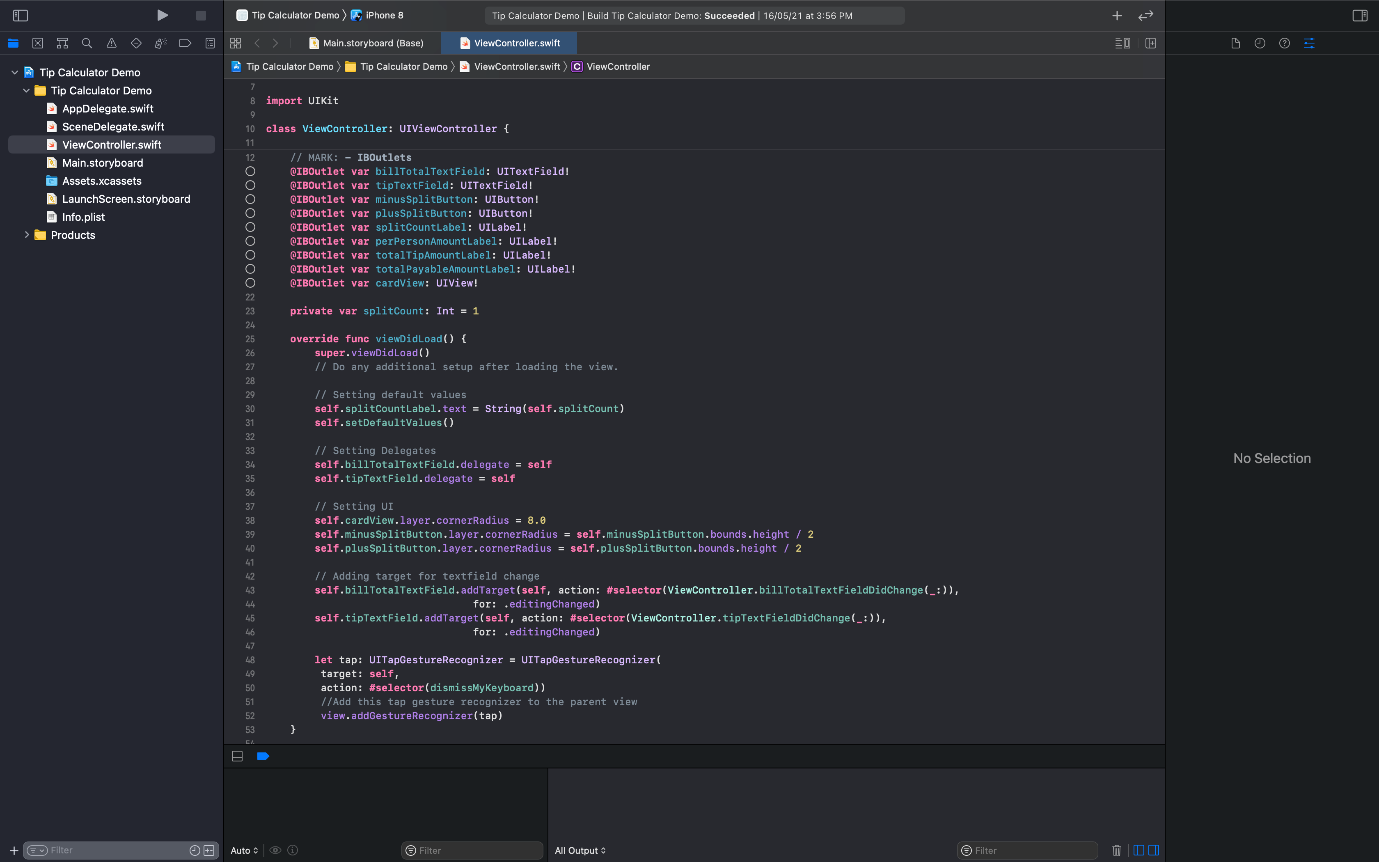
You should have your UI ready like this.



Now open assistant and make connections like below. The names below are self-explanatory.



First, we set the UI. We set the corner radius of the cardview, minusSplitButton and plusSplitButton. Then we set delegates and add target to the textfield, such that on filling the textfield, the calculator starts working.



Now we need to write the functions:

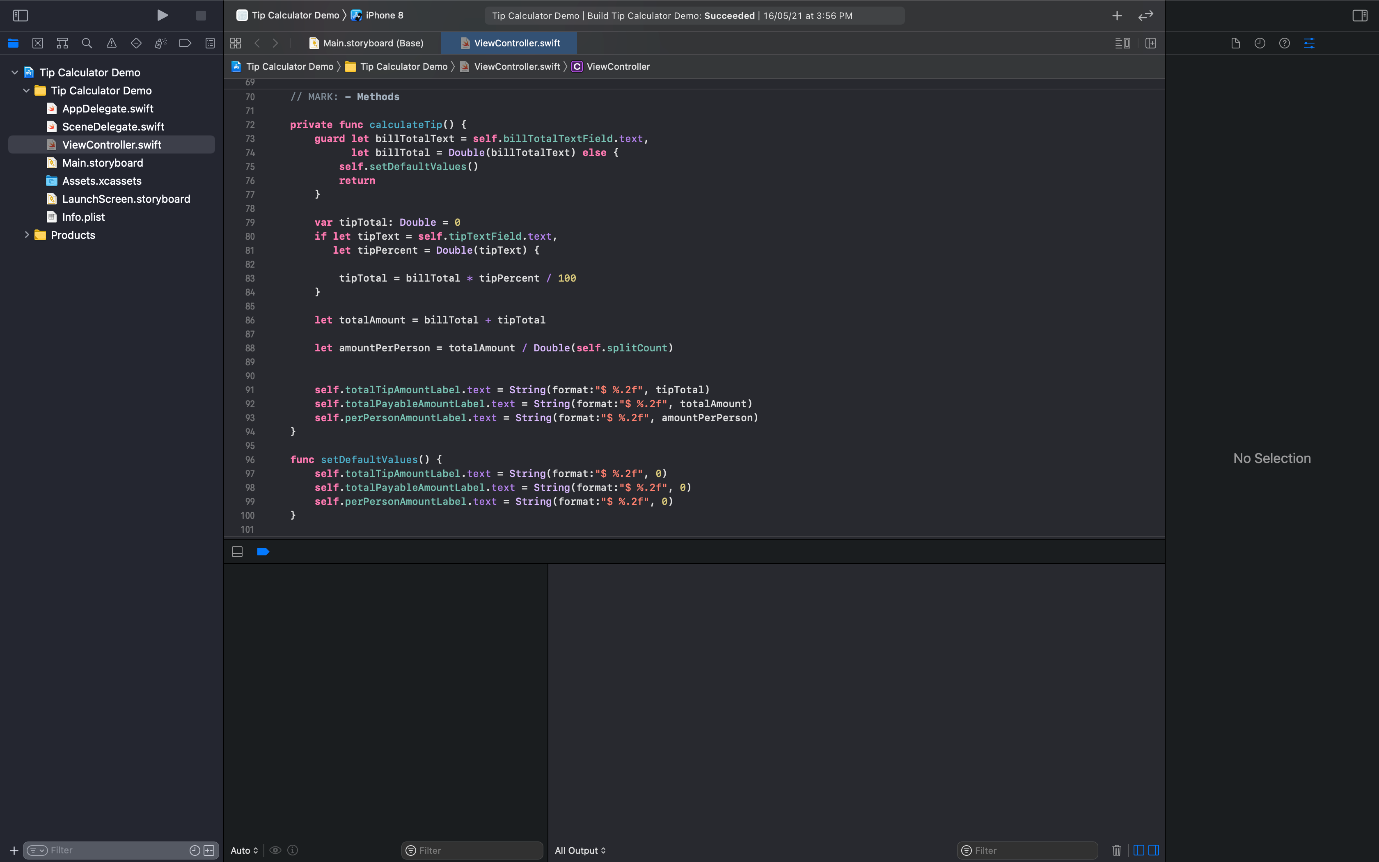
*@objc func billTotalTextFieldDidChange(\_ textField: UITextField) {*

*Self.calculateTip()*

*}*

*Objc* depicts objective c code. Apple allows this interoperability. Repeat the above process for *tipTextFieldDidChange****.***

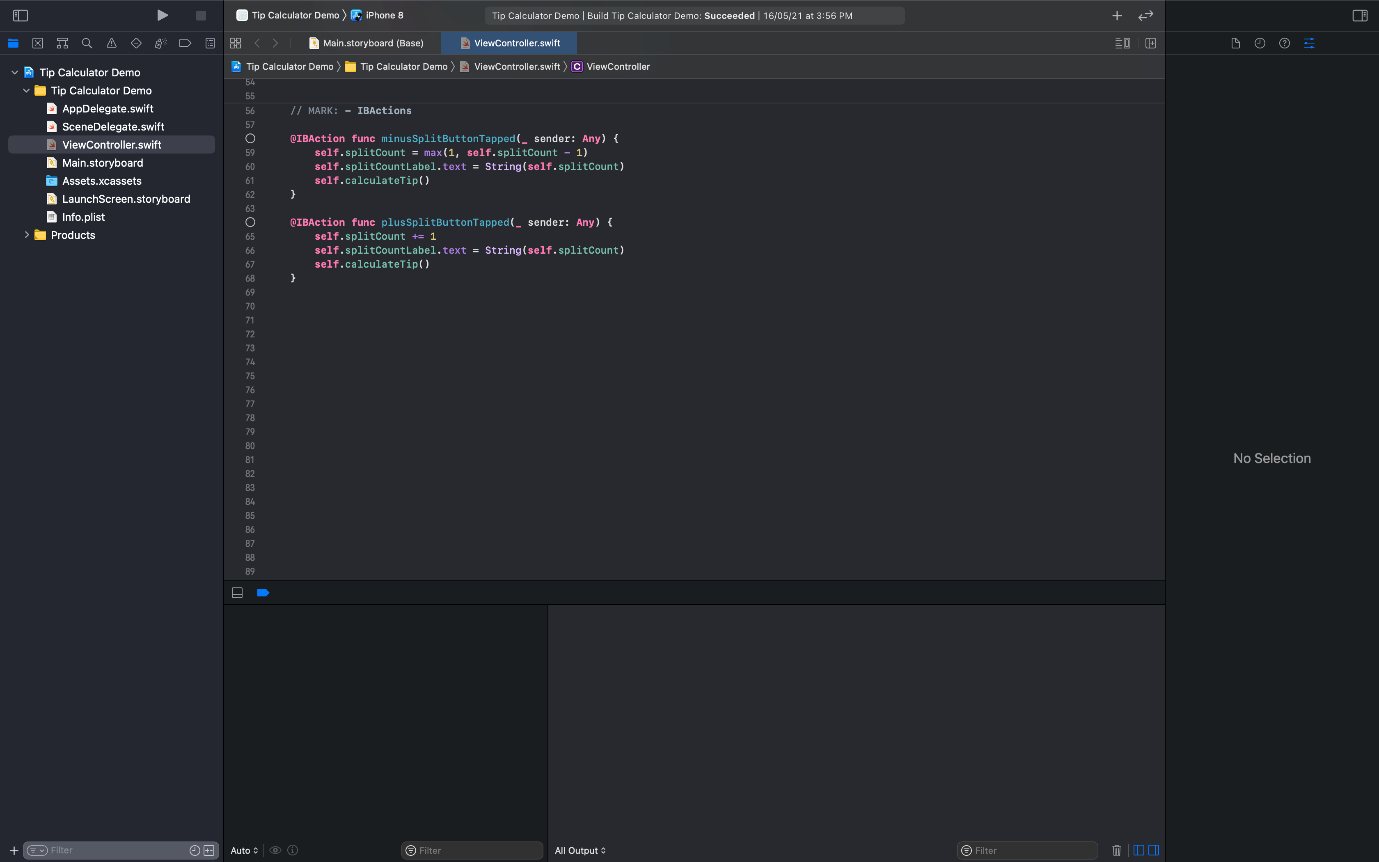
Then write the *calculateTip* function as below.



In this function we are taking the values set in *billTotalTextField* and in *tipTextField* and calculate the tip amount and the split amount. We then display the respective values in *totalTipAmountLabel, totalPayableAmountLabel* and in *perPersonAmountLabel.*

Now write the *setDefaultValues()* function. This function is setting the default value in the three respective values. Now call this function in *viewDidLoad* as default values.

Now make IBAction outlets from both the minus and plus buttons and write the following code.



We had earlier initialized *splitCount = 1* in the beginning. While we set it to 1 by default while the minus button is tapped, we deduct the value by 1. However, it doesn’t go below 1 and hence we choose the maximum of 1 and the deducted value. We set the value and call *calculateTip().*

You are now good to go. Run your app and you have your first ever app Tip Calculator ready!!!

This is just for beginners.

Where to go from here?

Watch “Your First iOS and UIKit App” by Raywenderlich.