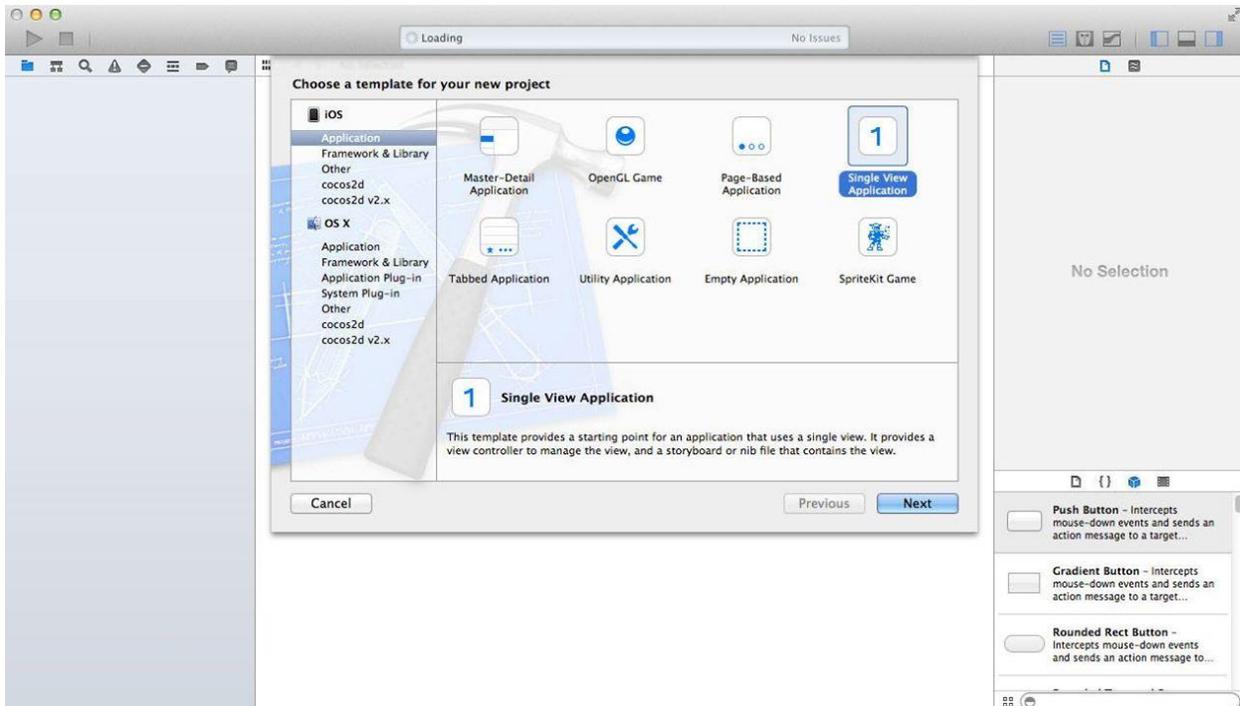
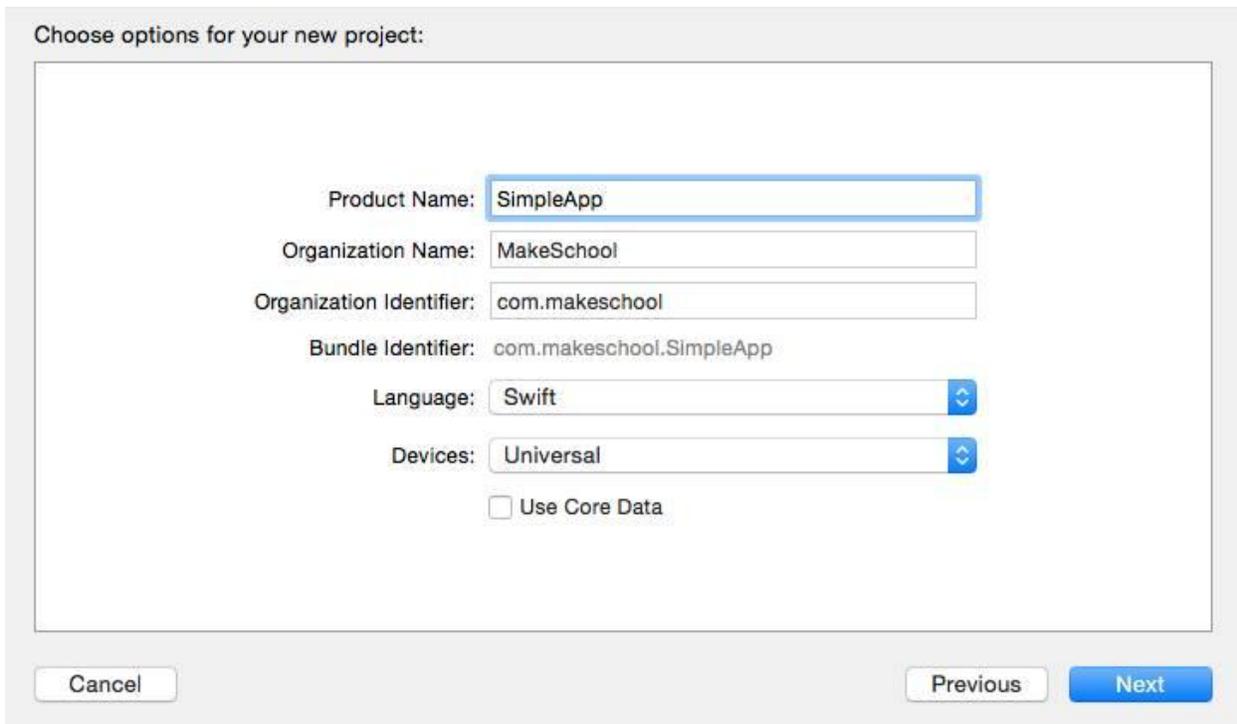


Dynamic Creation of Label, Image view, Button

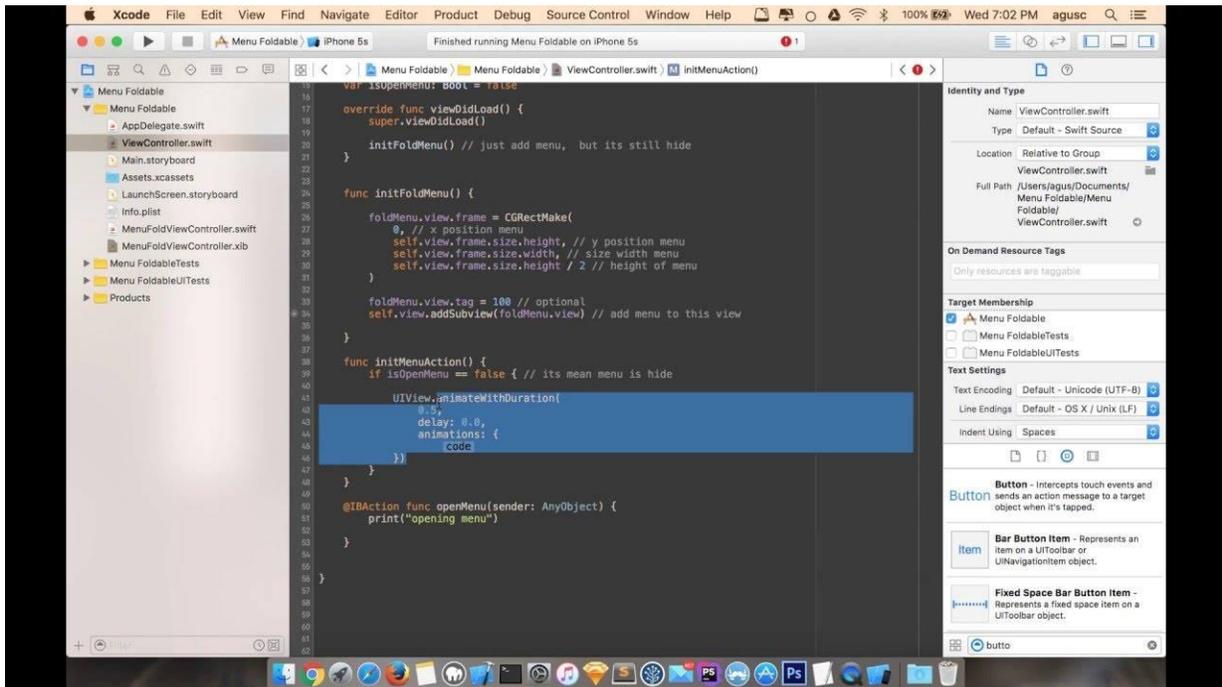
- Open Xcode and create new project.



- Give the product name for a project.



- Open the **ViewController.swift** and we have write the code here.



Dynamic Creation of Label

```
LET LABEL = UILabel(FRAME: CGRECT(X:100,,Y:80,WIDTH:100,HEIGHT:50))
    LABEL.TEXTALIGNMENT = .CENTER
    LABEL.TEXT="LABEL"
    SELF.VIEW.ADDSUBVIEW(LABEL)
```

- **Let** is used for declaring constant values that are not changed further in the coding.
- We declaring constant value for label as label.
- The label that is to be displayed is declared in Dynamic creation as **UILabel()**.
- The height, width, position of the text field is given as Core Graphics Rectangular(**CGRect**).
- Set an alignment for the label. And give the text for label to show in simulator.
- Self.view.addSubview(label) is used to run the entered code in the simulator, without this code the coding won't work.

Dynamic creation of Button

```
LET BUTTON = UIButton(FRAME: CGRECT(X: 100, Y: 150, WIDTH: 100, HEIGHT: 50))
    BUTTON.BACKGROUND_COLOR = .BLACK
    BUTTON.SETTITLE("BUTTON", FOR: .NORMAL)
    SELF.VIEW.ADDSUBVIEW(BUTTON)
```

- **Let** is used for declaring constant values that are not changed further in the coding.
- We declaring constant value for button as button.
- The button that is to be displayed is declared in Dynamic creation as **UIButton()**.
- The height, width, position of the text field is given as Core Graphics Rectangular(**CGRect**).
- Set the background color for the button.
- `Self.view.addSubview(button)` is used to run the entered code in the simulator, without this code the coding won't work.

Dynamic Creation of ImageView

- Var is used for declaring variable that can be changed further in the coding.
- We declare variable for ImageView as imageView.
- The button that is to be displayed is declared in Dynamic creation as **UIButton()**.
- The height, width, position of the text field is given as Core Graphics Rectangular(**CGRect**).
- Set the image for ImageView to show in simulator.
- `self.view.addSubview(imageView)` is used to run the entered code in the simulator, without this code the coding won't work.

Run and execution of simulator

- After finishing the coding part check for any error and fix the error and RUN the program by follow:
- The run button runs the program and select the iPhone Simulator and RUN the program.
- After that the simulator will open and It will show the dynamic creation of label, button, imageView