**Mobile Computing – iOS Spring’23**

**Assignment03**

**20 Points**

**This assignment is for building a calculator app, which will do the basic/intermediate mathematical operations. Your app should contain at least the following mentioned buttons:**

* **Numbers from 0-9**
* **AC to clear all and BS to clear the last character (i.e., backspace)**
* **Operands (+, -, ÷, x,** **% (i.e., percent), ln, x!, 10x )**
* **Dot (.)**
* **Equals (=)**
* **Sign change button (+/-)**
* **Trigonometric Functions (sin, cos, and tan)**

**Please follow the following instructions to complete this assignment.**

1. Open Xcode from the launchpad of your Mac.
2. Click on create a new Xcode project. Select the iOS platform and click on the App template.
3. Click on next, which will prompt you to choose options for the project.
4. Provide product name as **LastnameCalculatorApp**, “**edu.northwest**” for organization identifier, “**Storyboard**” as interface and **Swift** as language.
5. Click on next and select an appropriate location to save your app and click on create. A project directory will be loaded.
6. From the project navigator click on “Main.storyboard” file, a blank mobile screen will be loaded, where the required components for the app are added.

**The View**

Table 1: UI elements configuration

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 1 UILabel element | To display result | mathExpressionLBL |
| 23 UIButton elements | AC - To clear the result | clearExpression: |
| BS – Backspace | backspace: |
| (+/-) – To flip sign.  Ex: If the number is 3, on clicking this button the output should be -3 and vice versa. | flipSign: |
| (%) – To calculate percentage | percent: |
| ln – To calculate natural log.  Ex: | naturalLog: |
| (X!) To calculate factorial of anumber  Ex: 10! | factorial: |
| (10ˣ) – To compute square root | tenPow: |
| (sin) – to calculate the trigonometric sin value | calcSin: |
| (cos) – to calculate the trigonometric cos value | calcCos: |
| (tan) – to calculate the trigonometric tan value | calcTan: |
| (1/x) to get the inverse of a number | inverse: |
| (**÷**) – To compute integer division | tappedChar: |
| (x) – To compute multiplication | tappedChar: |
| (-) – To compute subtraction | tappedChar: |
| (+) – To compute addition | tappedChar: |
| (=) – To evaluate a mathematical expression | result: |
| (0-9) – Total 10 buttons for digits 0 to 9, respectively | tappedChar: |
| () – Dot button to add decimal values | tappedChar: |

*Note: Names that are ending with a colon (****:****) are actions.*

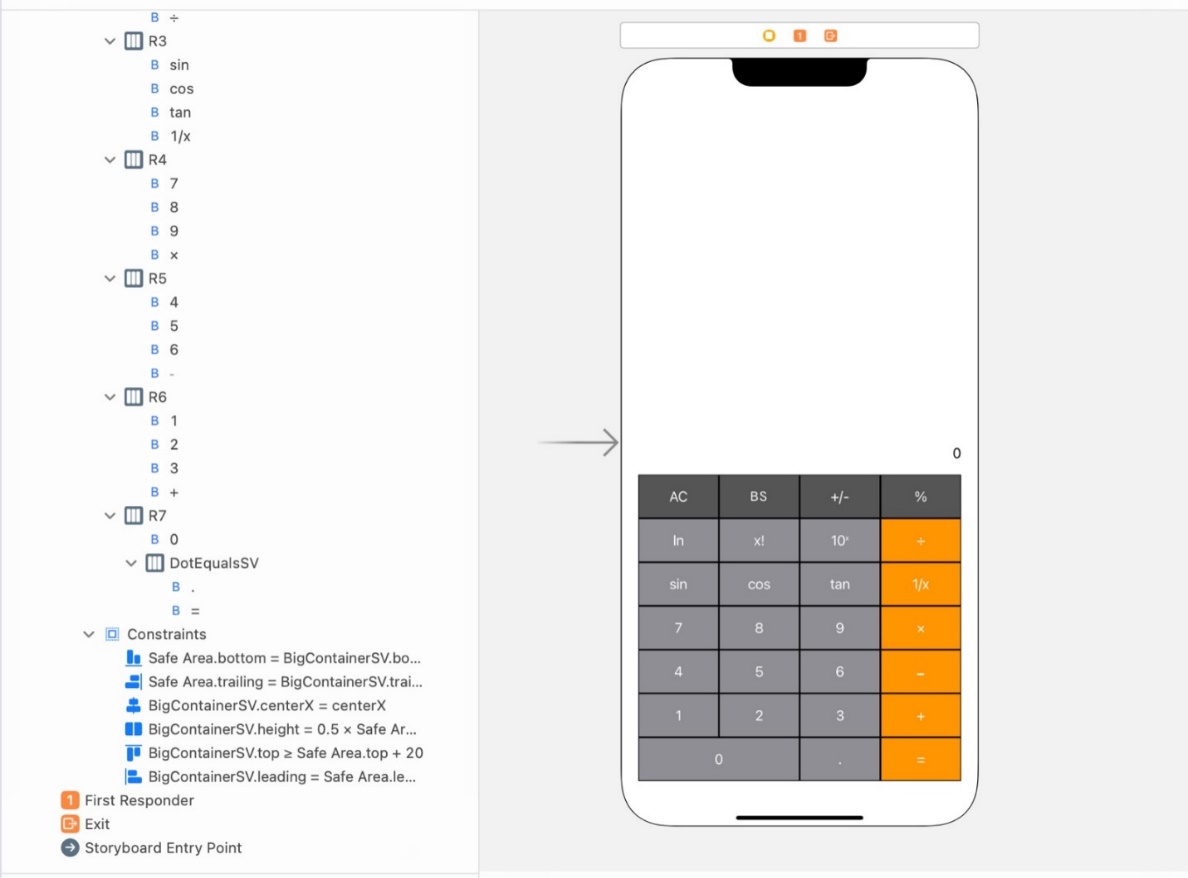
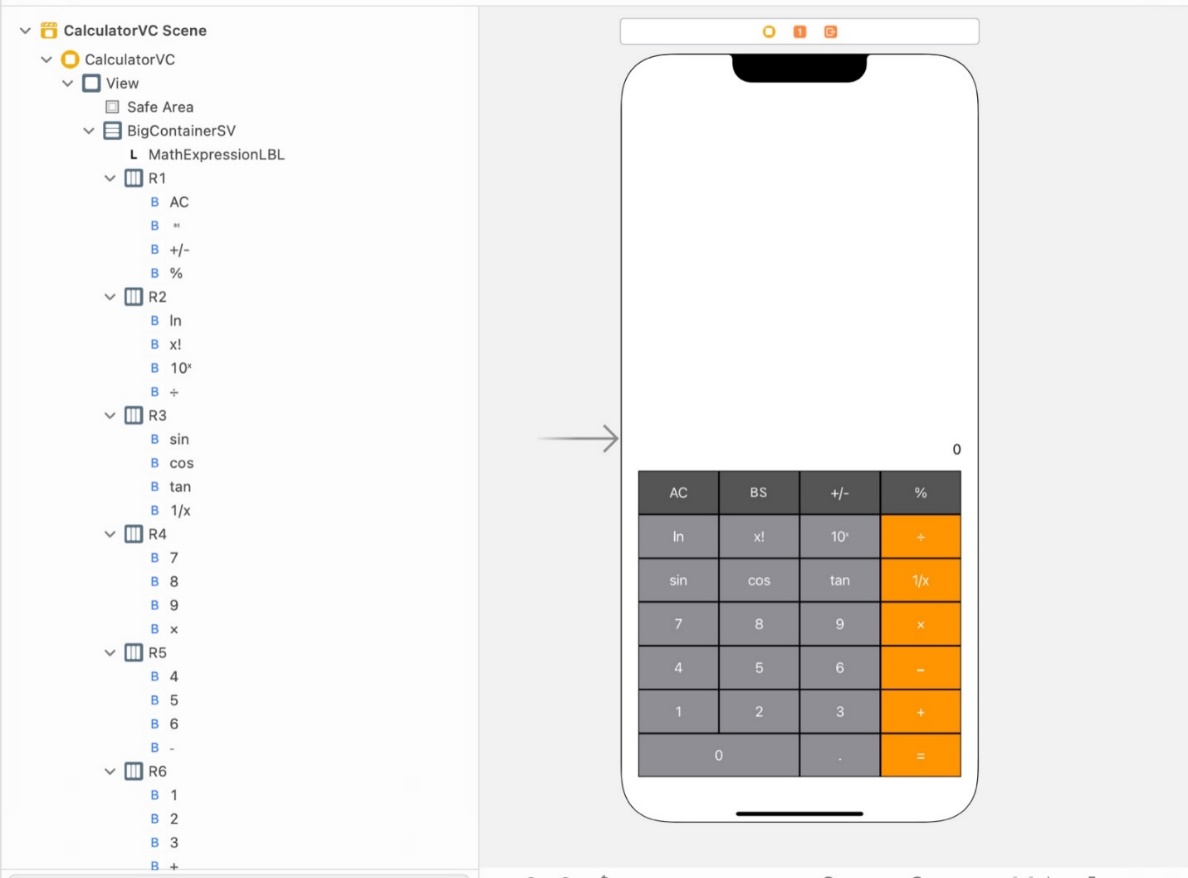
1. Open library (cmd+shift+l). Search for “label”, add (drag and drop) it to the storyboard.

Graphical user interface, text, application

Description automatically generated

1. Similarly, add 23 buttons to the storyboard from the library and fill in the colors respectively as shown in the picture.
   1. For **AC, BS, +/-,** and **%** buttons, fill color is Dark Gray Color.
   2. For **, , -, +,**1/x and **=** buttons, fill color is System Orange Color.
   3. Remaining buttons should have fill color set to System Gray Color.
   4. All the buttons should have stroke color set to Label Color.
   5. In Xcode, look Edit > Emoji & Symbols to find special Unicode characters.

Figure : Auto Layout



1. Now, all the required UI elements for the app are added to the storyboard. Apply auto layout to the app by adding constraints to UI elements as shown in the below figures.
   1. Hint: Follow the view hierarchy in the Figure 1 (i.e., document outline on the left side) to add and organize stack views and embed elements inside them.

Table 1 CalculatorVC’s properties for stack views

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stack View** | **Axis** | **Alignment** | **Distribution** | **Spacing** |
| BigContainerSV | Vertical | Fill | Fill | 0 |
| R1 | Horizontal | Fill | Fill Equally | 0 |
| R2 | Horizontal | Fill | Fill Equally | 0 |
| R3 | Horizontal | Fill | Fill Equally | 0 |
| R4 | Horizontal | Fill | Fill Equally | 0 |
| R5 | Horizontal | Fill | Fill Equally | 0 |
| R6 | Horizontal | Fill | Fill Equally | 0 |
| R7 | Horizontal | Fill | Fill Equally | 0 |
| DotEqualsSV | Horizontal | Fill | Fill | 0 |

**The Controller**

1. Create a Cocoa Touch Class “CalculatorVC” that is a sub class of UIViewController and assign it as class to Main.storyboard file.
2. Create outlet/action items as specified in Table 1.
3. The mathExpressionLBL is initially set to an empty string. Its number of lines is set to 0.
4. Create actions for all the buttons with type as UIButton.
   1. Note: Buttons 0 to 9, +, -, x, **÷** and will be connected to a single action (tappedChar:).
5. Action tappedChar: will pick the tapped character (sender.titleLabel?.text) and assign it to the mathExpressionLBL.
   1. It is important to understand how to append and evaluate expression correctly to achieve the chaining of calculations.
6. Action factorial: will calaulate the factorial of a number and assign it to mathExpressionLBL.
7. Action tenPow: will calaulate the 10th power of the number in the mathExpressionLBL and assign it to mathExpressionLBL.
8. Action calcSin: will calculate the trigonometric sin value of a number and assign it to mathExpressionLBL.
9. Action Inverse: will calculate the inverse of a number and assign it to mathExpressionLBL.
10. Action calcCos: will calculate the trigonometric cos value of a number and assign it to mathExpressionLBL.
11. Action calcTan: will calculate the trigonometric tan value of a number and assign it to mathExpressionLBL.
12. Action naturalLog: will calculate and assign it to the mathExpressionLBL.
13. Action percent: will calculate the percentage of a value and assign it to the mathExpressionLBL.
14. Action flipSign: will flip the sign of a value and assign it to the mathExpressionLBL.
15. Action backspace: will remove the last character of the mathExpressionLBL.
16. Action result: will evaluate the mathExpressionLBL and update it with the result.

*Note: You may add the following package to your Xcode project to evaluate mathematical expressions.*

* [*https://github.com/peredaniel/MathExpression.git*](https://github.com/peredaniel/MathExpression.git)

**Your app should pass the following test cases to consider it as a successful implementation.**

1. **20 + 600 = 620**
2. **Tap AC to clear everything.**
3. **20 (+/-) + 5 = -15**
4. **10 ÷ 3 ln = 1.20397**
5. **5 ÷ 0 = inf or Error**
6. **10! sin = 0.2639**
7. **2** **(10ˣ) (1/x) = 0.01**
8. **.3 x 4 - .2 = 1**
9. **0 cos = 1**
10. **5 tan = -3.3805**
11. **Chaining of calculations**

* **3 + 4 \* => 7 \* 5 - => 35 – 10 ÷ => 25 ÷ 5 + => 5 + 0 = 5**
* **30+4(+/-) => -34/17 => 2() => 0.01(1/x) => 100**

**Sample Output (Video Link):**

<https://app.vidgrid.com/view/aUR7l6jacPHM>