7SENG002W Week 1 SwiftUI Tutorial

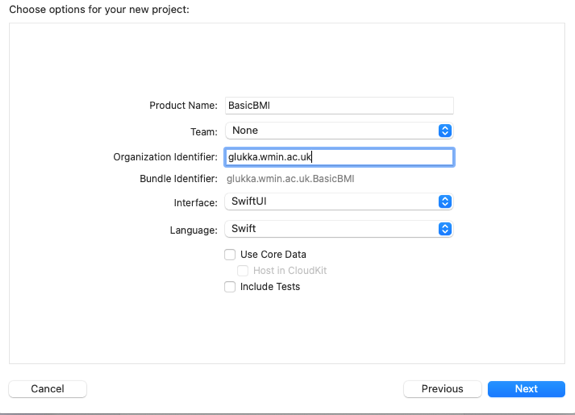
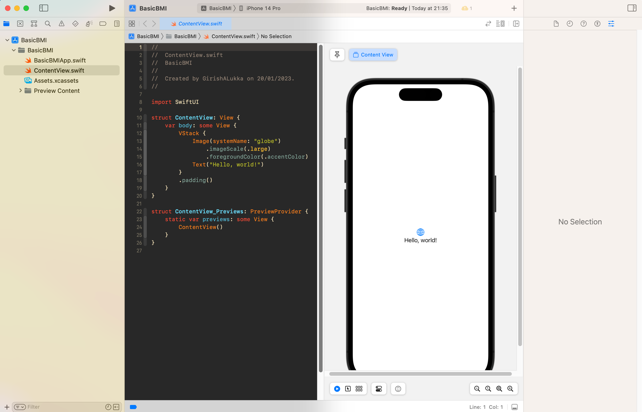
To get familiar with SwiftUI application development and XCode 14 let us build a basic BMI Calculator. The requirements for this app are simple – get user height, weight, calculate BMI and display it with a suitable message.

Launch XCode and select iOS App template and interact with pop-ups to create a BasicBMI (Body Mass Index) app as shown below:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated



Create a label using Text view to display “BMI Calculator Ver1”

Create two TextField Views for weight and height inputs. There are other ways to get user inputs, however, we will use TexFields here. Can you think of how else might we get this?

Create a button with a label: “Calculate BMI”

Create a label to display the bmi value, formatting it 2 dp.

(You may have done this in Python, string interpolation)

Create a label to display a suitable message based on the value of bmi.

Embed all these in a VStack as shown below:

A screenshot of a computer program

Description automatically generated

Any input in the TextField is bound to the variable height, note use of @State and $ in TextField.

Modifiers applied to Text

Modifiers applied to TextfField

Code Explanation

TextField("Enter Weight (kg)",text: $weight)

The first argument to the TextField view is the placeholder that provides hints to the user on how to enter input. In this field, the user must enter weight in kilograms e.g., 85 (meaning 85 kg).

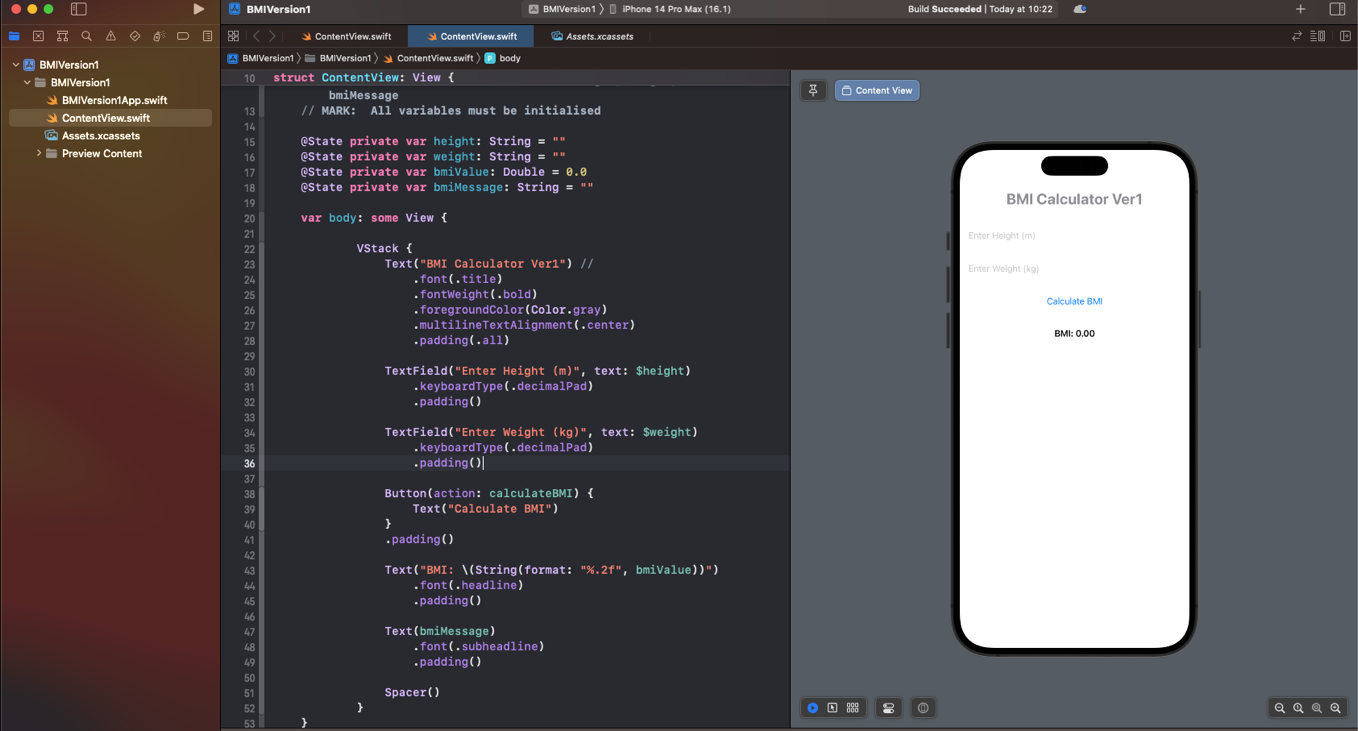
Two private state variables weight and height are bound to a

TextField object through the text parameter.

TextField("Enter Height (m)",text: $height)

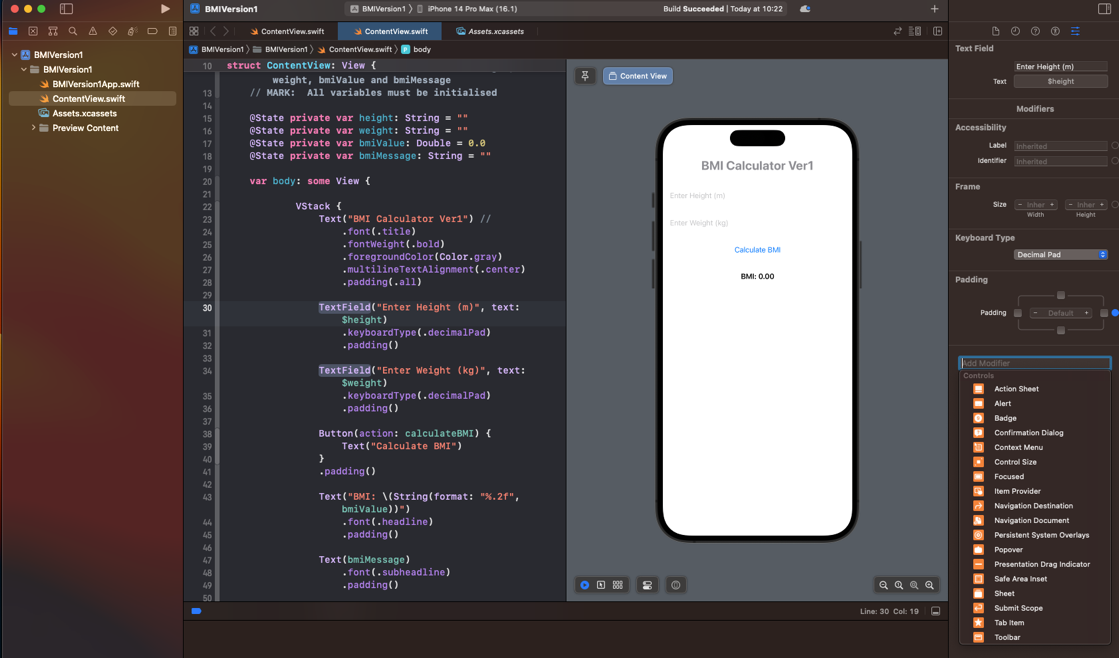
TextField("Enter Weight (kg)",text: $weight)

Note use $ prefix to bind the state variable to the view, as shown in the code snippet.

Preview updates dynamically - the view within XCode so there is no need to use a simulator.

Examine the effect of commenting out the TextField modifier relating KeyboardType.

What other KeyboardType modifiers are available?

Highlight the height TextField by hovering the mouse over and see the information in the properties panel. 

Click in the Add Modifier tab and examine the options, try some of them later and see how the code changes as well as the preview.

A screen shot of a cell phone

Description automatically generated A cell phone with a screen

Description automatically generated

Just for demo here, I have added border to height TextField.

Next, the button “Calculate BMI” has to be managed. As it is, there is no action attached to it.

List the events and actions that must take place for the button to be effective.

1. Check that both height and weight have data i.e. neither is empty
2. The height and weight entered numeric and greater than 0 and cast them as double – why?
3. If both 1 and 2 are true, calculate bmi.
4. Assign bmi to bmiValue.
5. Create a suitable message based on bmi value and assign that to bmiMessage.
6. What should happen if either of 1 or 2 is false?
7. Examine the func calculateBMI(){ ……..} and see if you agree with it?

Button Code:

A computer screen shot of a program

Description automatically generated

Complete code with code folding:

A screen shot of a computer

Description automatically generated

Test your code in actual simulator – iPhone 14 Pro Max

Version 1 of BMI Calculator is complete.

Make a copy of the whole project folder and save it in a new folder.

Rename the copied folder BMIVersion2 and use that to continue working on the app.

This how I have it on my computer:

A screenshot of a computer

Description automatically generatedTest your A screenshot of a computer

Description automatically generated

## Version 2

Add a background color to the app – this can be done using ZStack as the first view with modifier applied to ZStack as shown below:

A screenshot of a computer program

Description automatically generated

Next add some more Text Views beside the Textfields and build a UI as shown below using additional modifiers:

A screenshot of a calculator

Description automatically generated

Next use ternary function with foregroundColor modifier to set the button color to gray if there is no data in the textfields, otherwise set it to blue.

A screenshot of a calculator

Description automatically generated A screenshot of a calculator

Description automatically generated

Disable the CalculateBMI button whilst there is no data in the textfields.

A screenshot of a calculator

Description automatically generated A screenshot of a calculator

Description automatically generated A screenshot of a calculator

Description automatically generated

Create an alert pop-up when non numeric data has been entered telling the user.

Add a variable :

@State **var** showAlert::Bool = **false**

Add alert modifier to VStack as shown:

A screen shot of a computer code

Description automatically generated

Modify the func calculateBMI() {……..} to activate alert.

This is end of version 2.

Same as before, make a copy, etc.

## Version 3 – A more challenging set of requirements to be implemented.

Requirements:

Keep a record of BMI calculations that includes date.

Show all bmi records as a list with each row showing the date, bmivalue and percentage change from last time, if it exists.

Date can be set by user but only up to current date, no future dates.

Display a percentage change in bmi value from previous time, if there is a previous record.

A reminder – last year in your OOP module, you did similar work for the GUI application.

Below are the app images to help you figure out what data structures and processing you will need to do. See if you can find errors.

This version does not require persistence.

A screenshot of a cell phone

Description automatically generated A screenshot of a calendar

Description automatically generated A screenshot of a calculator

Description automatically generated A screenshot of a cell phone

Description automatically generated

This will be fully covered in week2 tutorial, for now carry out the analysis of what must happen rather than how.