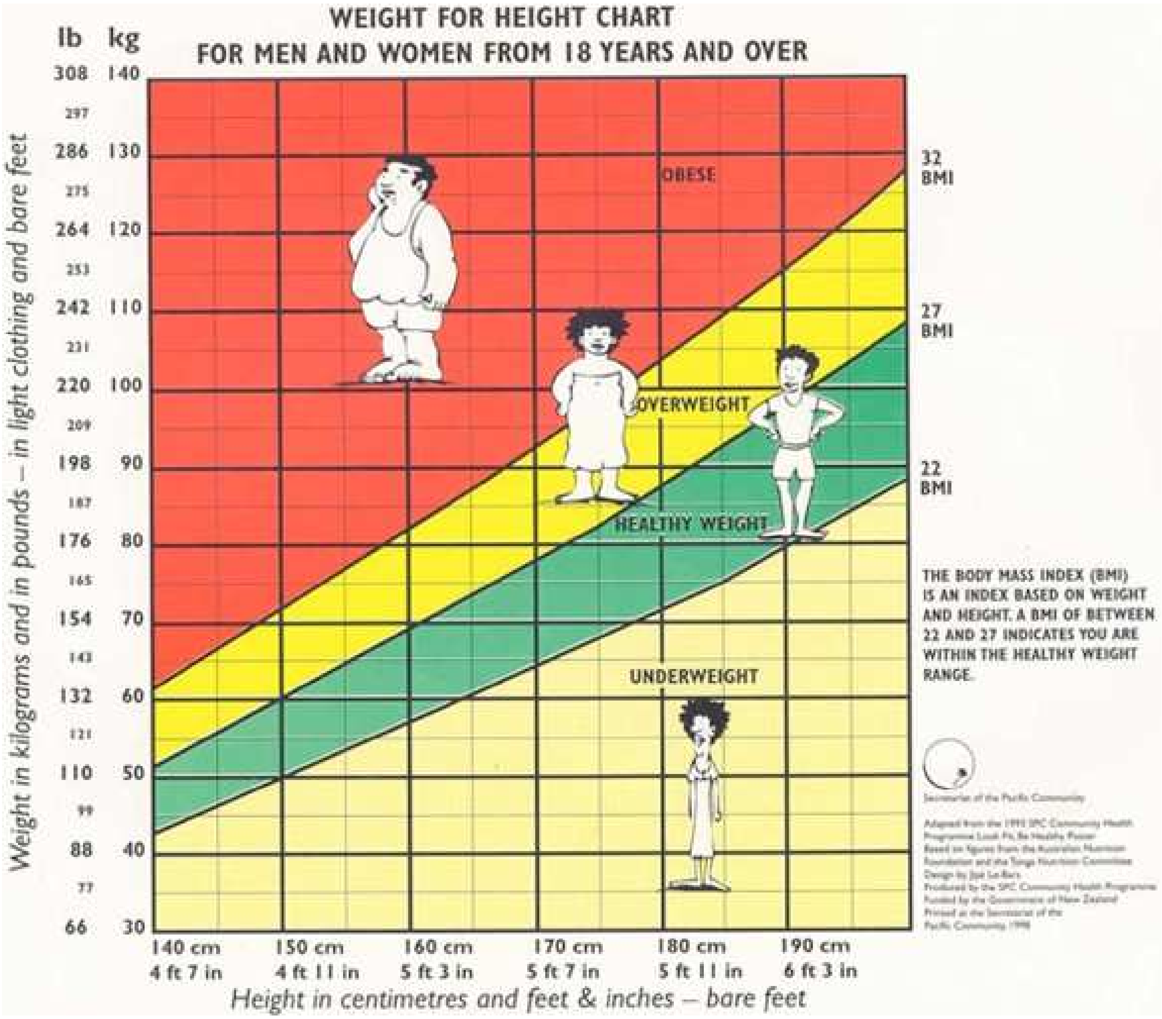
React Native

Keyboard Input, State, Timer

BMI Calculator App



WEIGHT FOR HEIGHT CHART
FOR MEN AND WOMEN FROM 18 YEARS AND OVER

THE BODY MASS INDEX (BMI) IS AN INDEX BASED ON WEIGHT AND HEIGHT. A BMI OF BETWEEN 22 AND 27 INDICATES YOU ARE WITHIN THE HEALTHY WEIGHT RANGE.

Secretariat of the Pacific Community

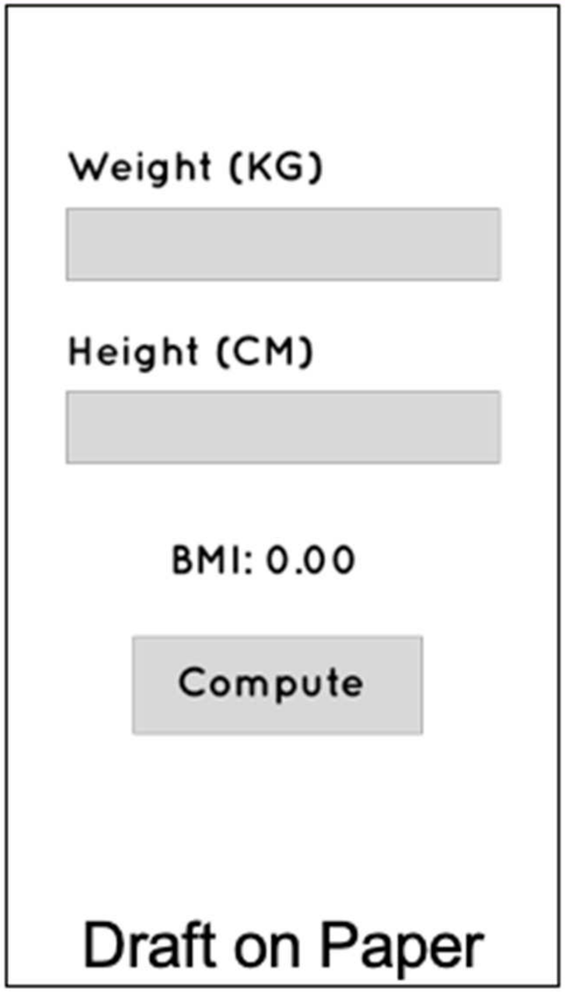
Adapted from the 1995 SPC Community Health
Programme Look Fit, Be Healthy Poster
Based on figures from the Australian Nutrition
Foundation and the Tonga Nutrition Committee
Design by Jipé Le-Bars
Produced by the SPC Community Health Programme
Funded by the Government of New Zealand
Printed at the Secretariat of the
Pacific Community 1998

HIENLTH

Making BMI Calculator App

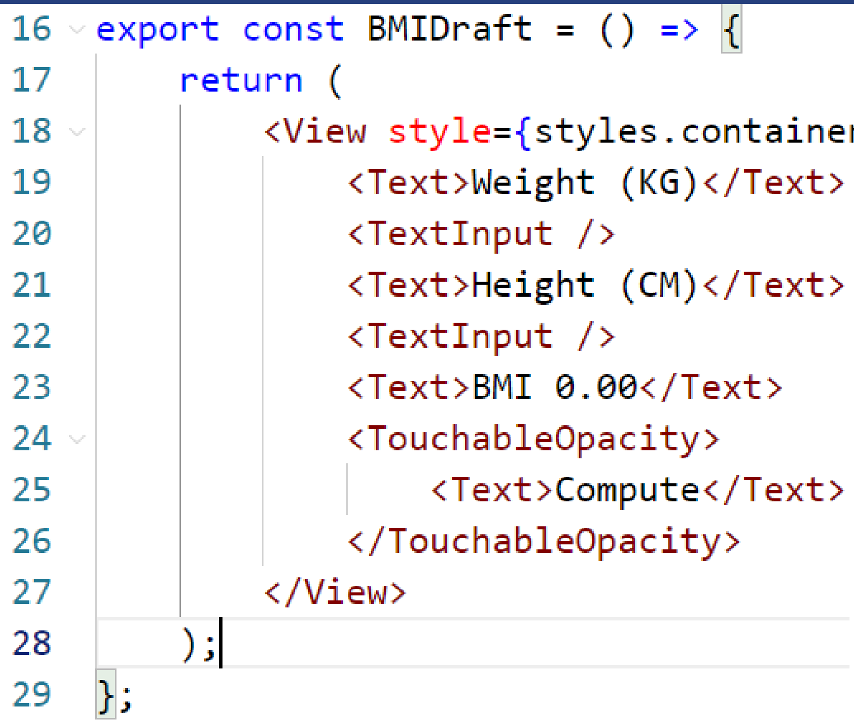
1. Create container that fill whole screen
2. Set the flexDirection of container to ‘column’ and its justifyContent: ‘center’
3. Start Adding Elements
   1. Text
   2. TextInput
   3. TouchableOpacity

- 20 minutes ==> TO DO

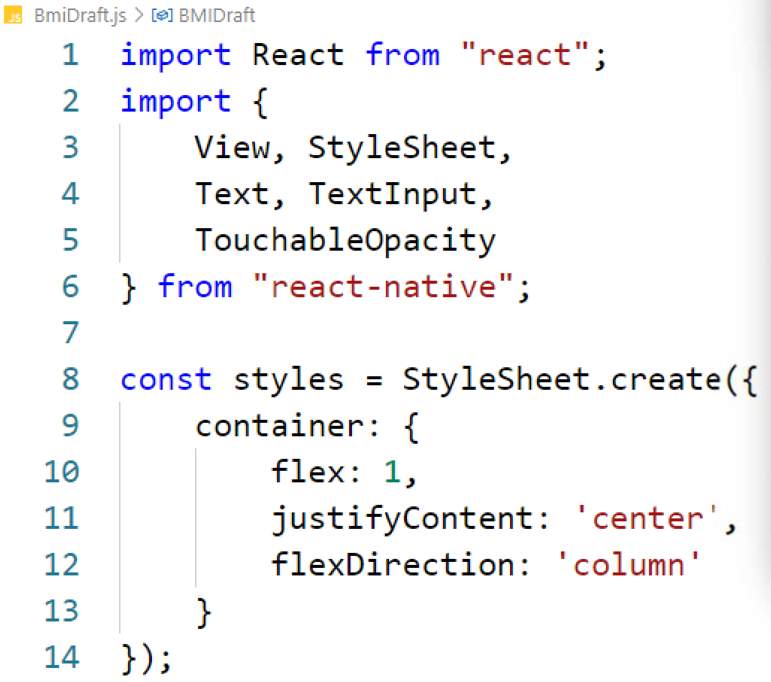


Draft on Paper

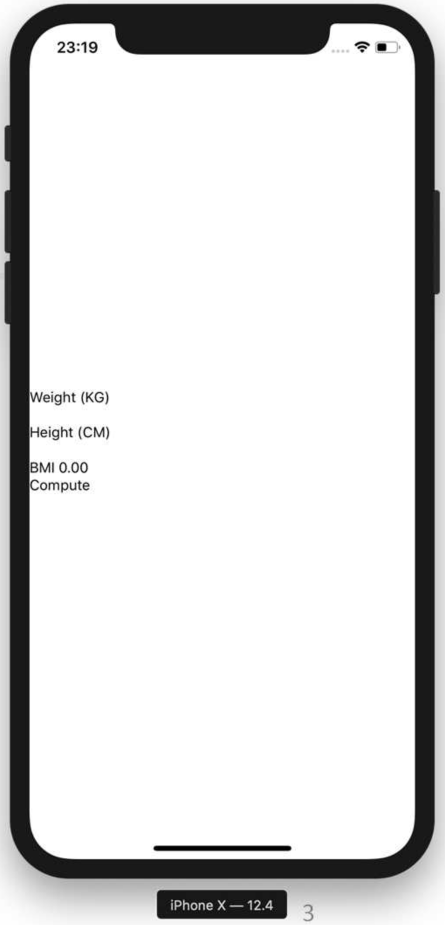
Making BMI Calculator App



```
export const BMIDraft = () => {
    return (
        <View style={styles.container}>
            <Text>Weight (KG)</Text>
            <TextInput />
            <Text>Height (CM)</Text>
            <TextInput />
            <Text>BMI 0.00</Text>
            <TouchableOpacity>
                <Text>Compute</Text>
            </TouchableOpacity>
        </View>
    );
};
```

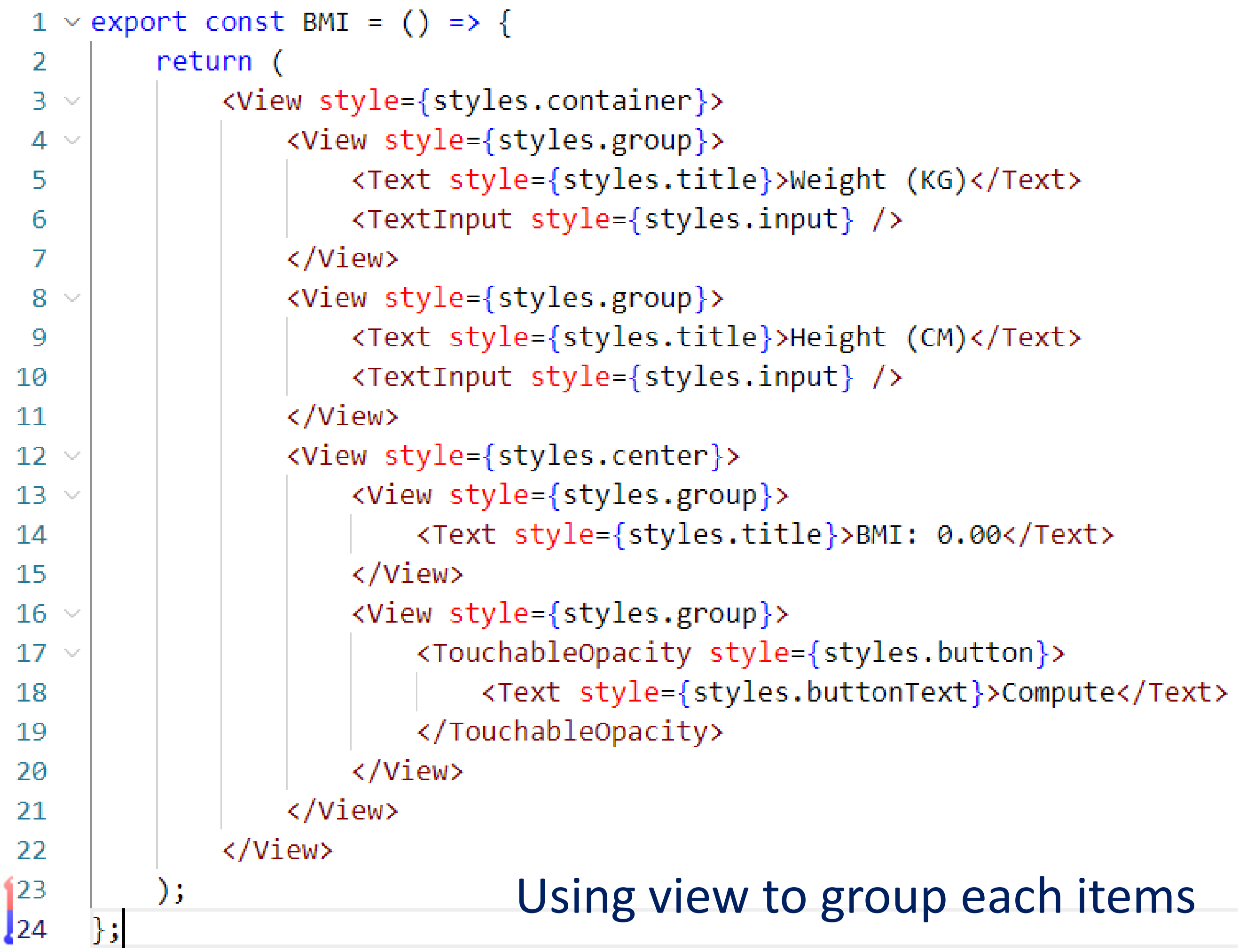


```
import React from "react";
import {
    View, StyleSheet,
    Text, TextInput,
    TouchableOpacity
} from "react-native";

const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'center',
        flexDirection: 'column'
    }
});
```

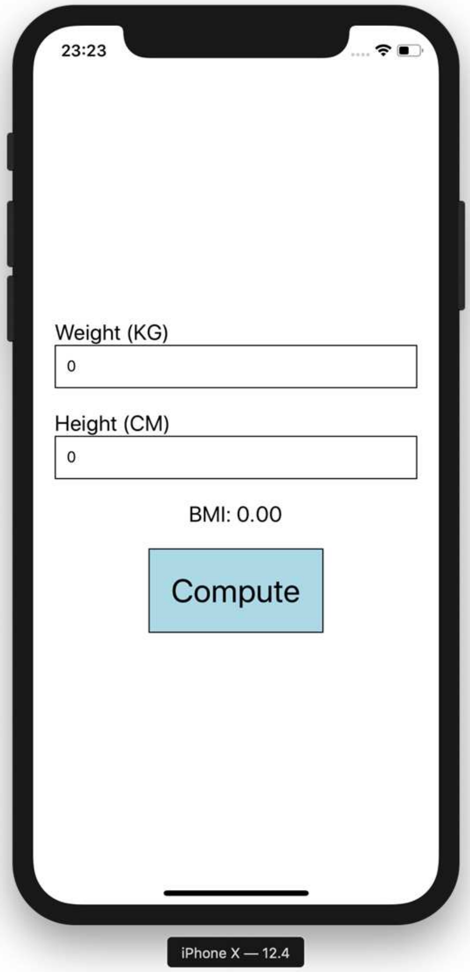


```
Weight (KG)
Height (CM)
BMI 0.00
Compute
```

Making BMI Calculator App

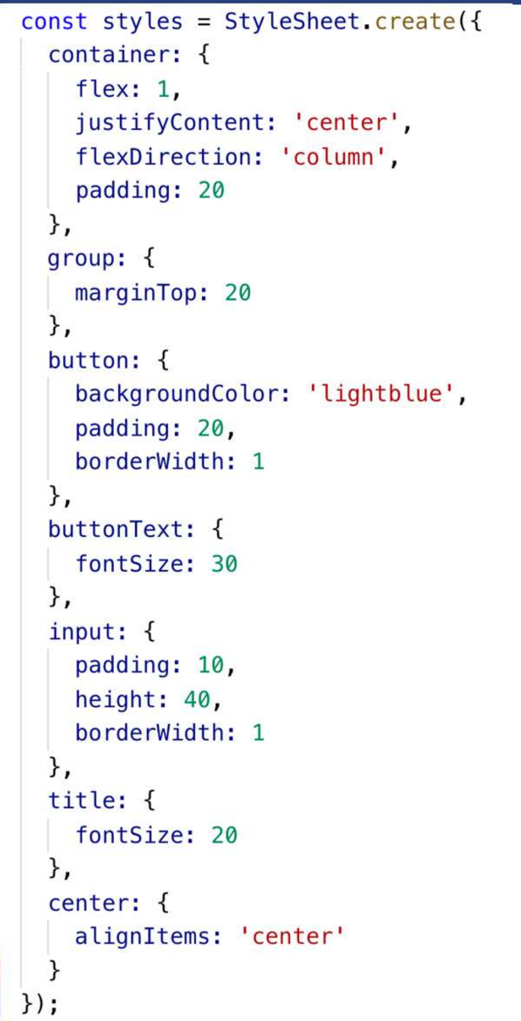


```
export const BMI = () => {
    return (
        <View style={styles.container}>
            <View style={styles.group}>
                <Text style={styles.title}>Weight (KG)</Text>
                <TextInput style={styles.input} />
            </View>
            <View style={styles.group}>
                <Text style={styles.title}>Height (CM)</Text>
                <TextInput style={styles.input} />
            </View>
            <View style={styles.center}>
                <View style={styles.group}>
                    <Text style={styles.title}>BMI: 0.00</Text>
                </View>
                <View style={styles.group}>
                    <TouchableOpacity style={styles.button}>
                        <Text style={styles.buttonText}>Compute</Text>
                    </TouchableOpacity>
                </View>
            </View>
        </View>
    );
};
```

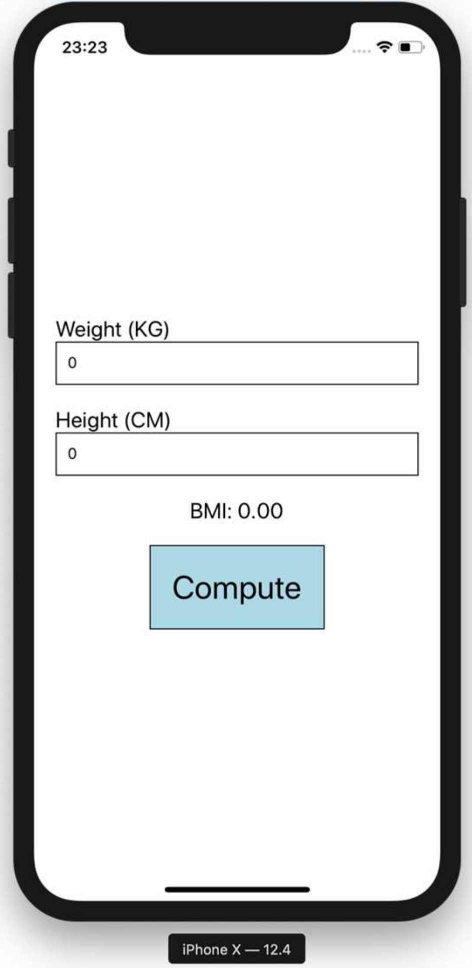


Using view to group each items

Making BMI Calculator App

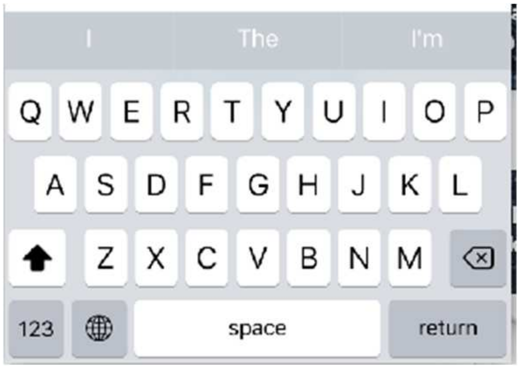


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    flexDirection: 'column',
    padding: 20
  },
  group: {
    marginTop: 20
  },
  button: {
    backgroundColor: 'lightblue',
    padding: 20,
    borderWidth: 1
  },
  buttonText: {
    fontSize: 30
  },
  input: {
    padding: 10,
    height: 40,
    borderWidth: 1
  },
  title: {
    fontSize: 20
  },
  center: {
    alignItems: 'center'
  }
});
```



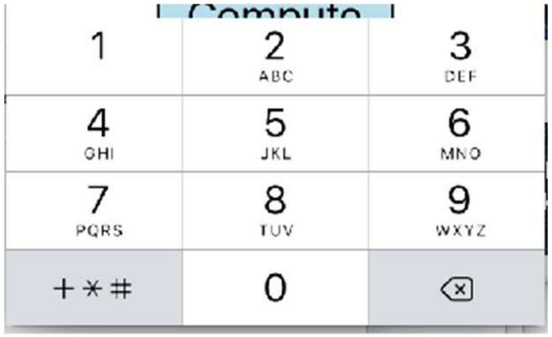
Keyboard Types (Cross-Platform)

```
<TextInput  keyboardType='default'>
```

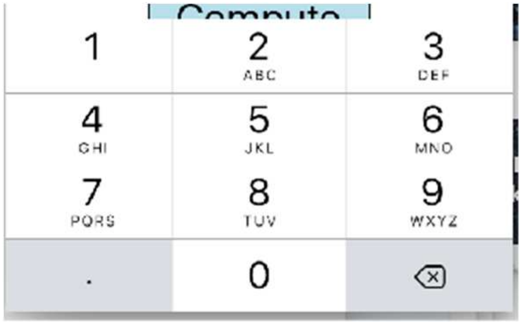


default

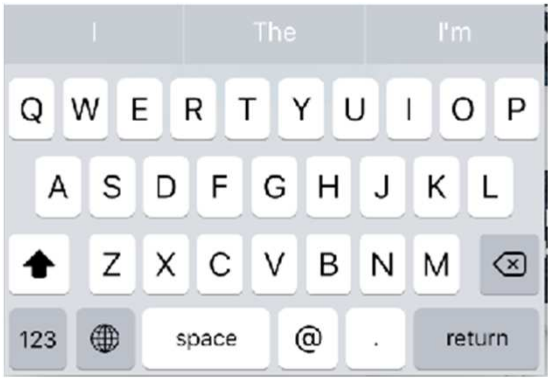
iOS (9.3)



phone-pad



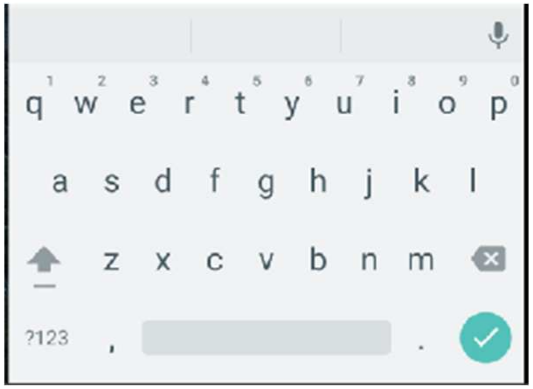
numberic



email-address

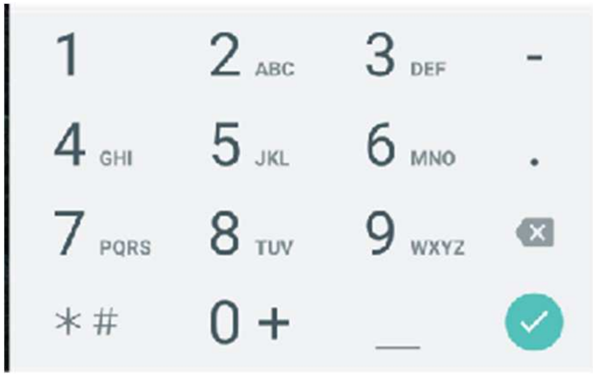
Keyboard Types (Cross-Platform)

```
<TextInput  keyboardType='default'>
```

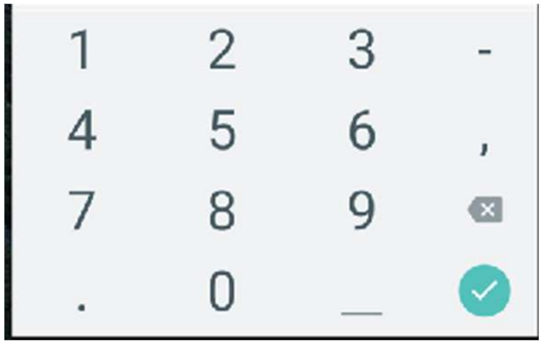


default

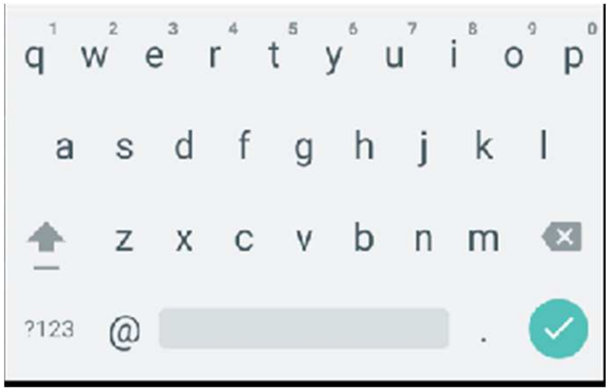
Android (9.3)



phone-pad



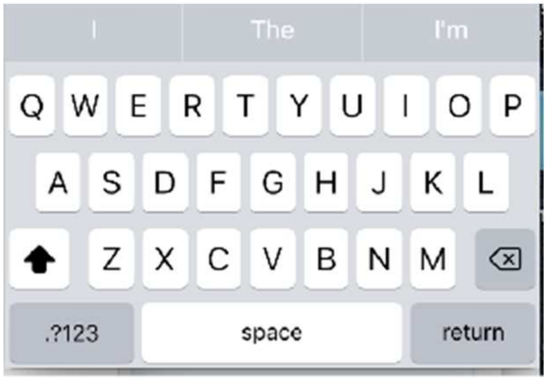
numberic



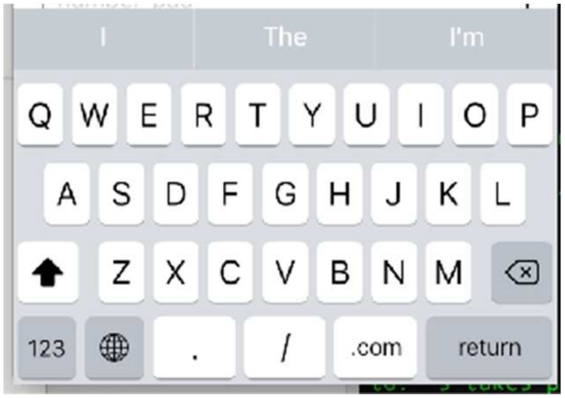
email-address

More Keyboard Types (iOS Only)

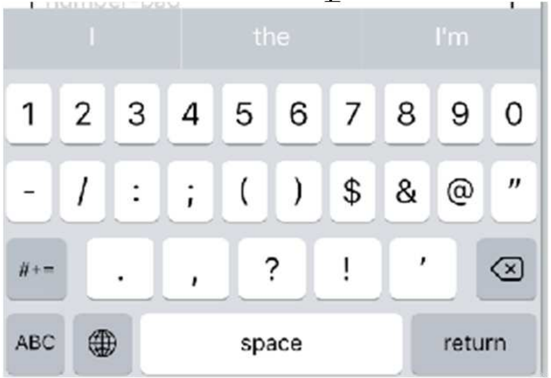
```
<TextInput  keyboardType='default'>
```



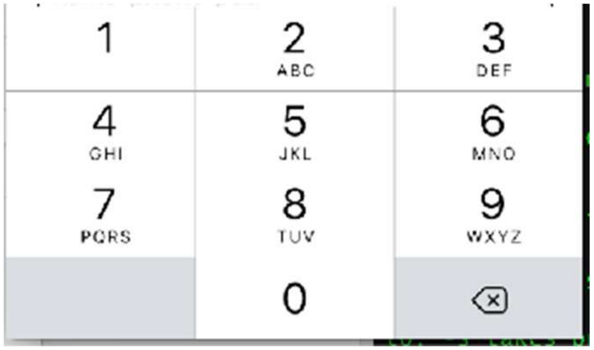
ascii-capable



url



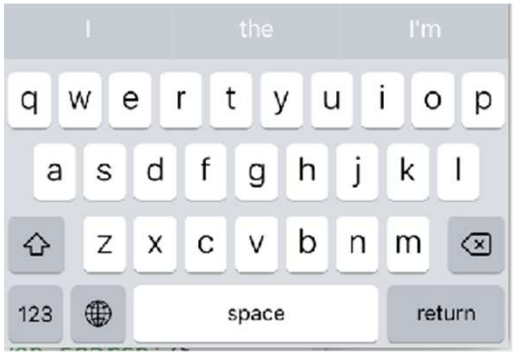
numbers-punctuation



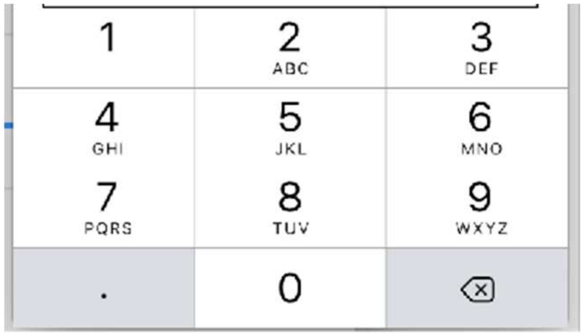
number-pad

More Keyboard Types (iOS Only)

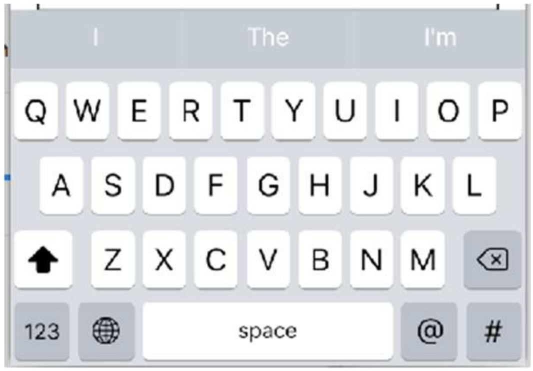
```
<TextInput  keyboardType=‘default’>
```



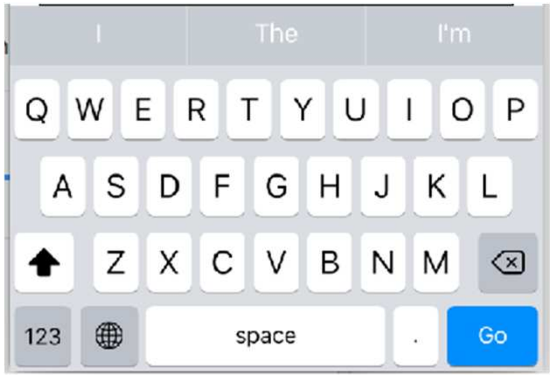
name-phone-pad



decimal-pad

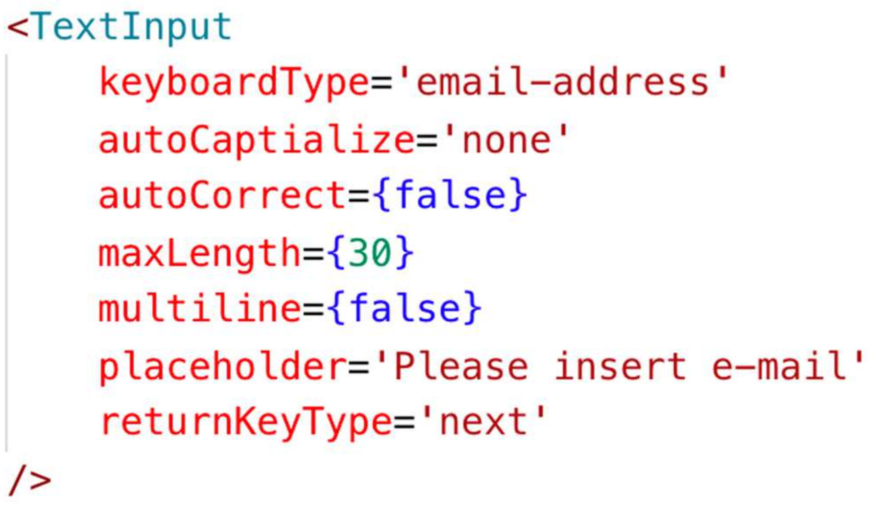


twitter

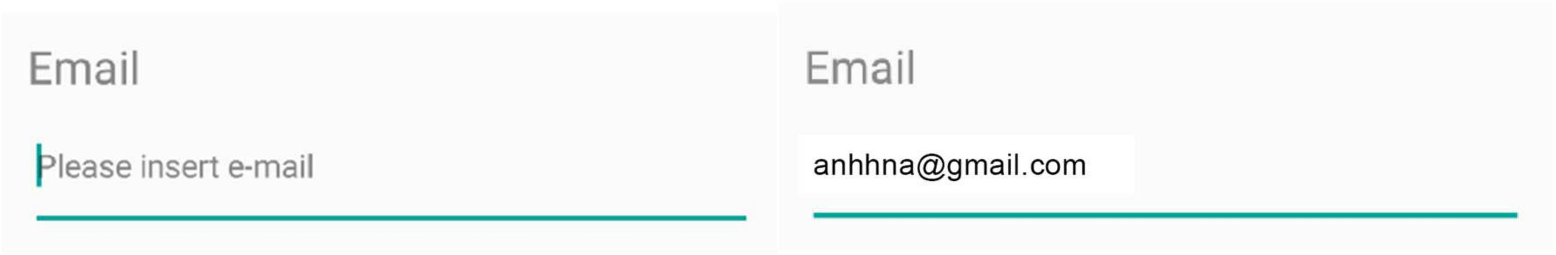


web-search

More TextInput Attribute



```
<TextInput
    keyboardType='email-address'
    autoCaptialize='none'
    autoCorrect={false}
    maxLength={30}
    multiline={false}
    placeholder='Please insert e-mail'
    returnKeyType='next'
/>
```



Email

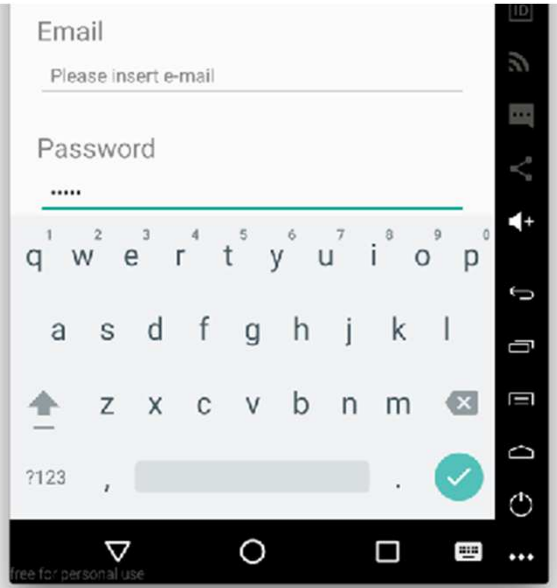
Please insert e-mail

Email

anhhna@gmail.com

Password TextInput

```
<TextInput
    keyboardType='default'
    secureTextEntry={true}
    ...
/>
```

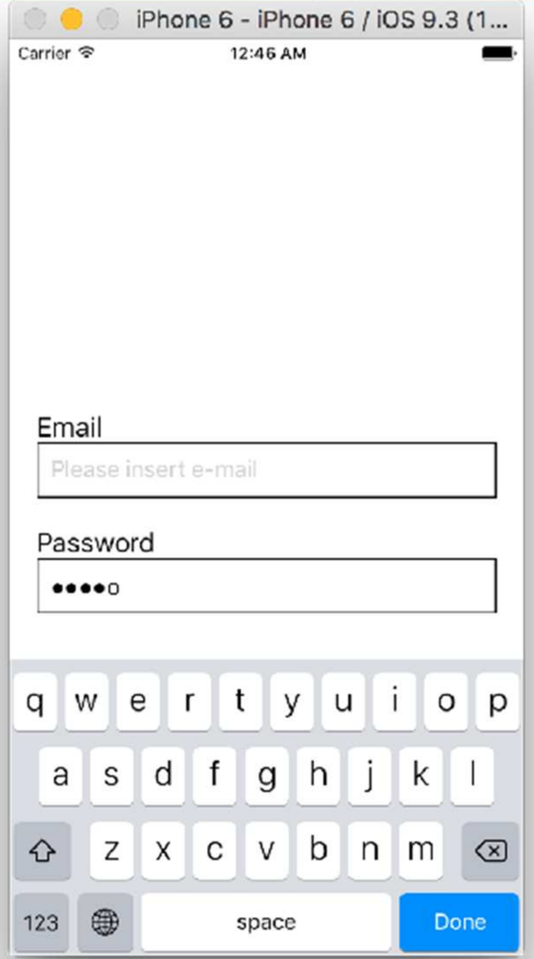


Email

Please insert e-mail

Password

•••••



iPhone 6 - iPhone 6 / iOS 9.3 (1...

Carrier 12:46 AM

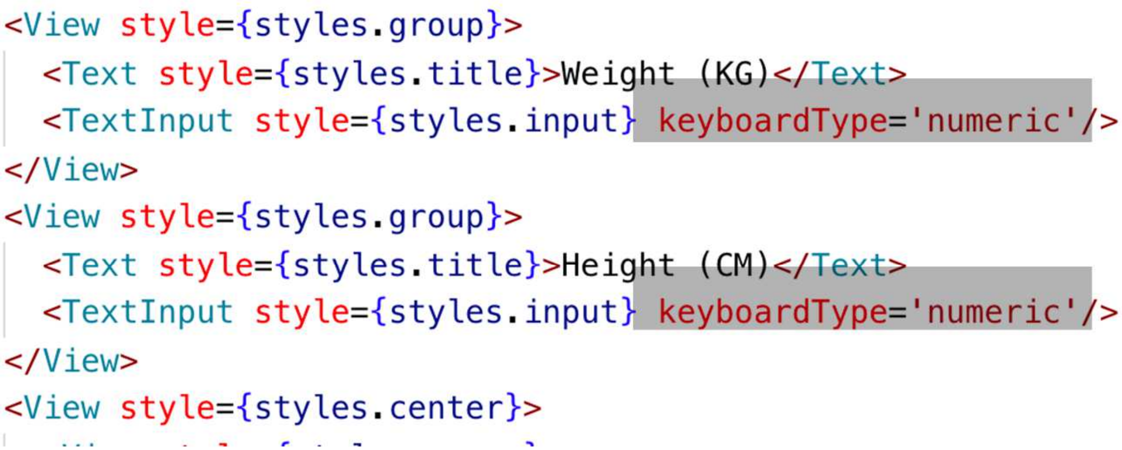
Email

Please insert e-mail

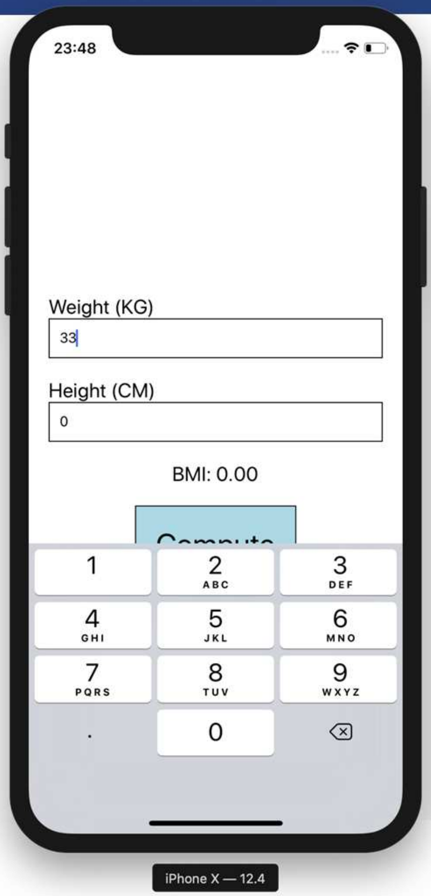
Password

••••o

KeyboardType



```
<View style={styles.group}>
  <Text style={styles.title}>Weight (KG)</Text>
  <TextInput style={styles.input} keyboardType='numeric'/>
</View>
<View style={styles.group}>
  <Text style={styles.title}>Height (CM)</Text>
  <TextInput style={styles.input} keyboardType='numeric'/>
</View>
<View style={styles.center}>
```



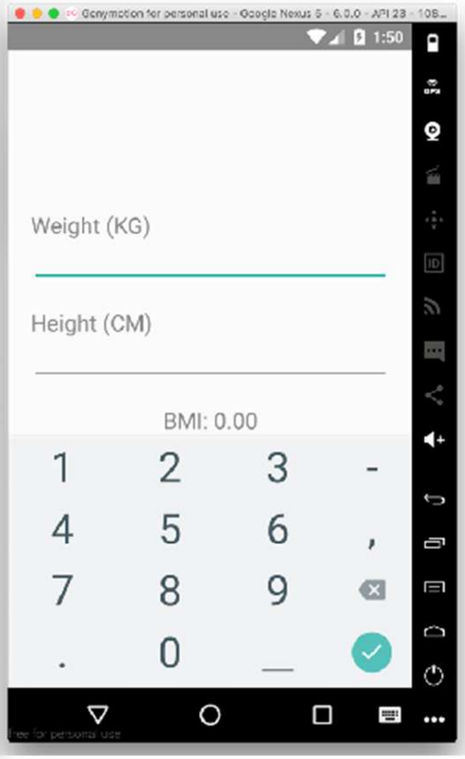
Weight (KG)

33

Height (CM)

0

BMI: 0.00



Weight (KG)

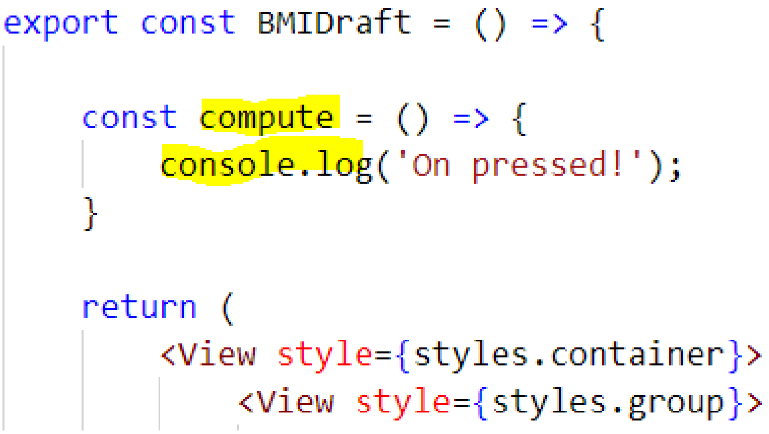
Height (CM)

BMI: 0.00

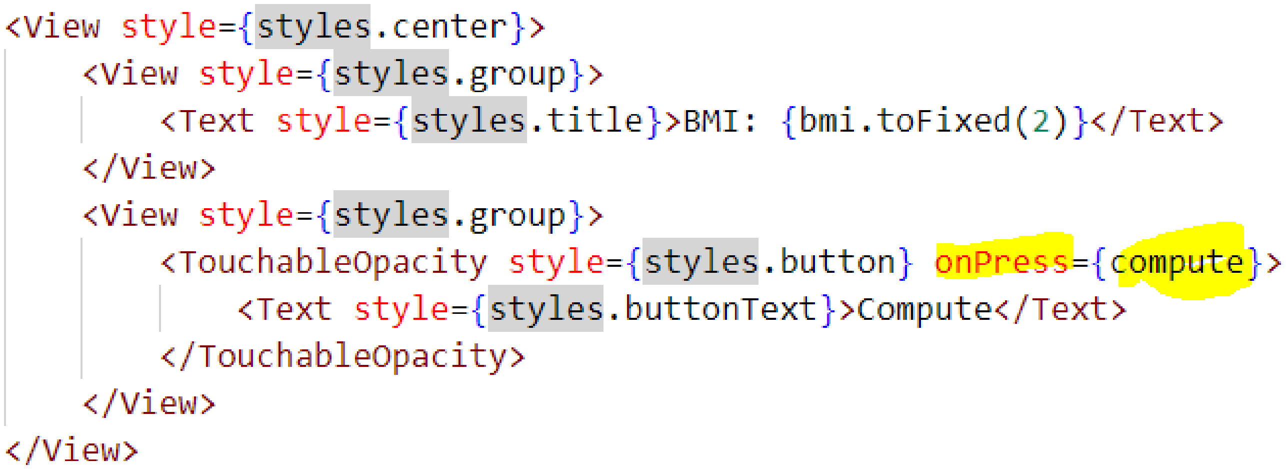
Genymotion for personal use - Google Nexus 5 - 6.0.0 - API 23 - 108...
1:50
1
2
3
-
4
5
6
,
7
8
9
.
0
_
free for personal use

iOS -> Cmd-K to Open/Hide Keyboard

Add User Input Feedback



```
export const BMIDraft = () => {

    const compute = () => {
        console.log('On pressed!');
    }

    return (
        <View style={styles.container}>
            <View style={styles.group}>
```



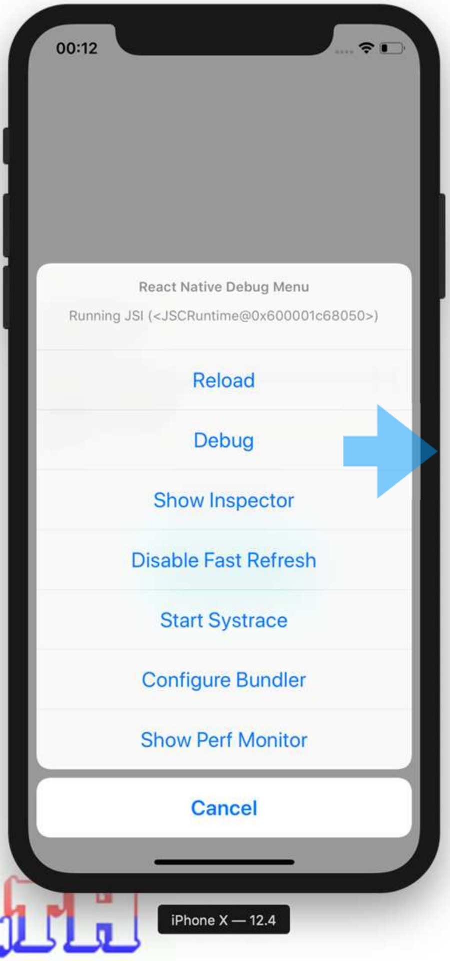
```
<View style={styles.center}>
    <View style={styles.group}>
        <Text style={styles.title}>BMI: {bmi.toFixed(2)}</Text>
    </View>
    <View style={styles.group}>
        <TouchableOpacity style={styles.button} onPress={compute}>
            <Text style={styles.buttonText}>Compute</Text>
        </TouchableOpacity>
    </View>
</View>
```

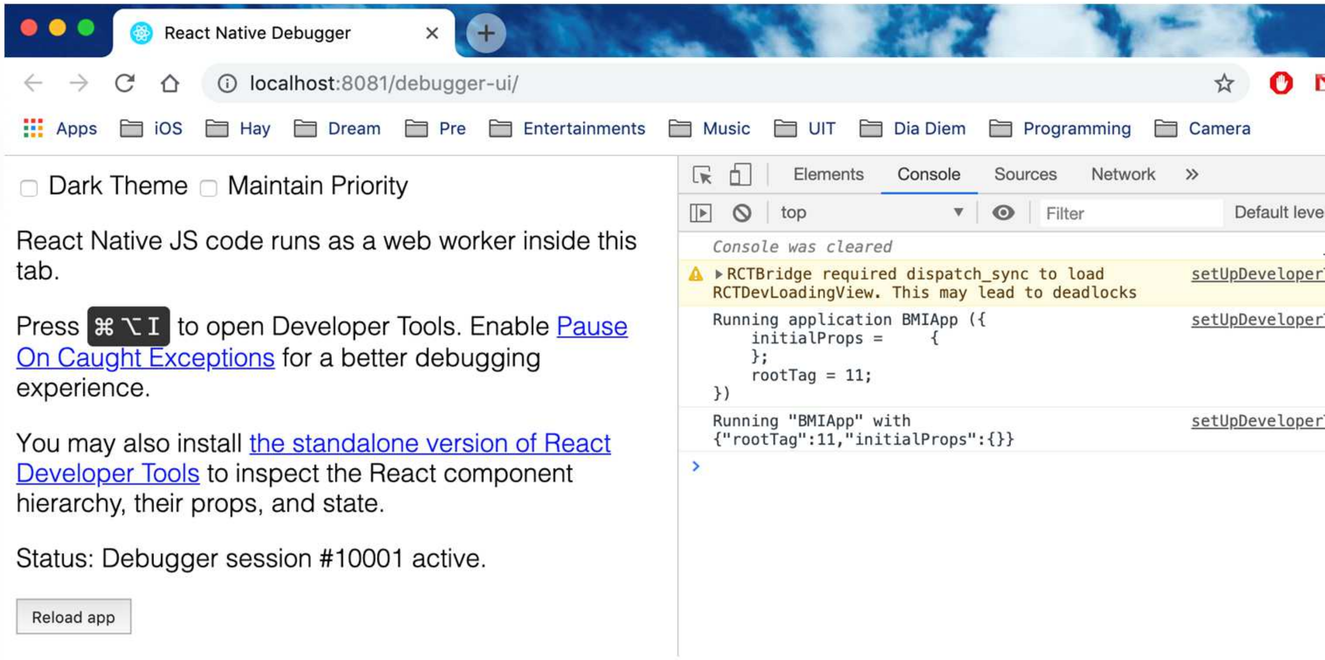
Enable JS Debugger

Cmd - D (iOS)

Menu Button

(Android)





□ Dark Theme □ Maintain Priority

React Native JS code runs as a web worker inside this tab.

Press ⌘⌥I to open Developer Tools. Enable Pause On Caught Exceptions for a better debugging experience.

You may also install the standalone version of React Developer Tools to inspect the React component hierarchy, their props, and state.

Status: Debugger session #10001 active.

Reload app

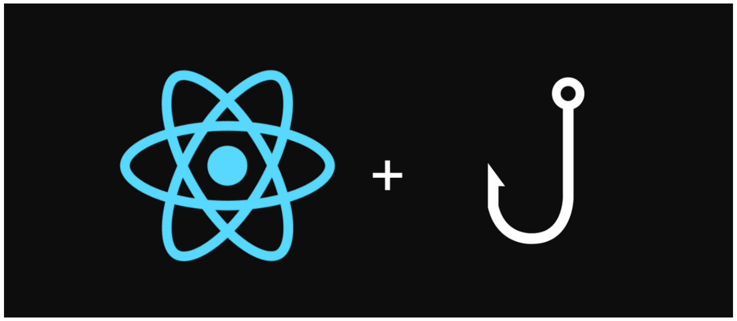
```
Console was cleared
⚠ ▸RCTBridge required dispatch_sync to load              setUpDeveloper
 RCTDevLoadingView. This may lead to deadlocks
 Running application BMIApp ({                          setUpDeveloper
     initialProps =     {
     };
     rootTag = 11;
 })
 Running "BMIApp" with                                  setUpDeveloper
 {"rootTag":11,"initialProps":{}}
>
```

State Management

State

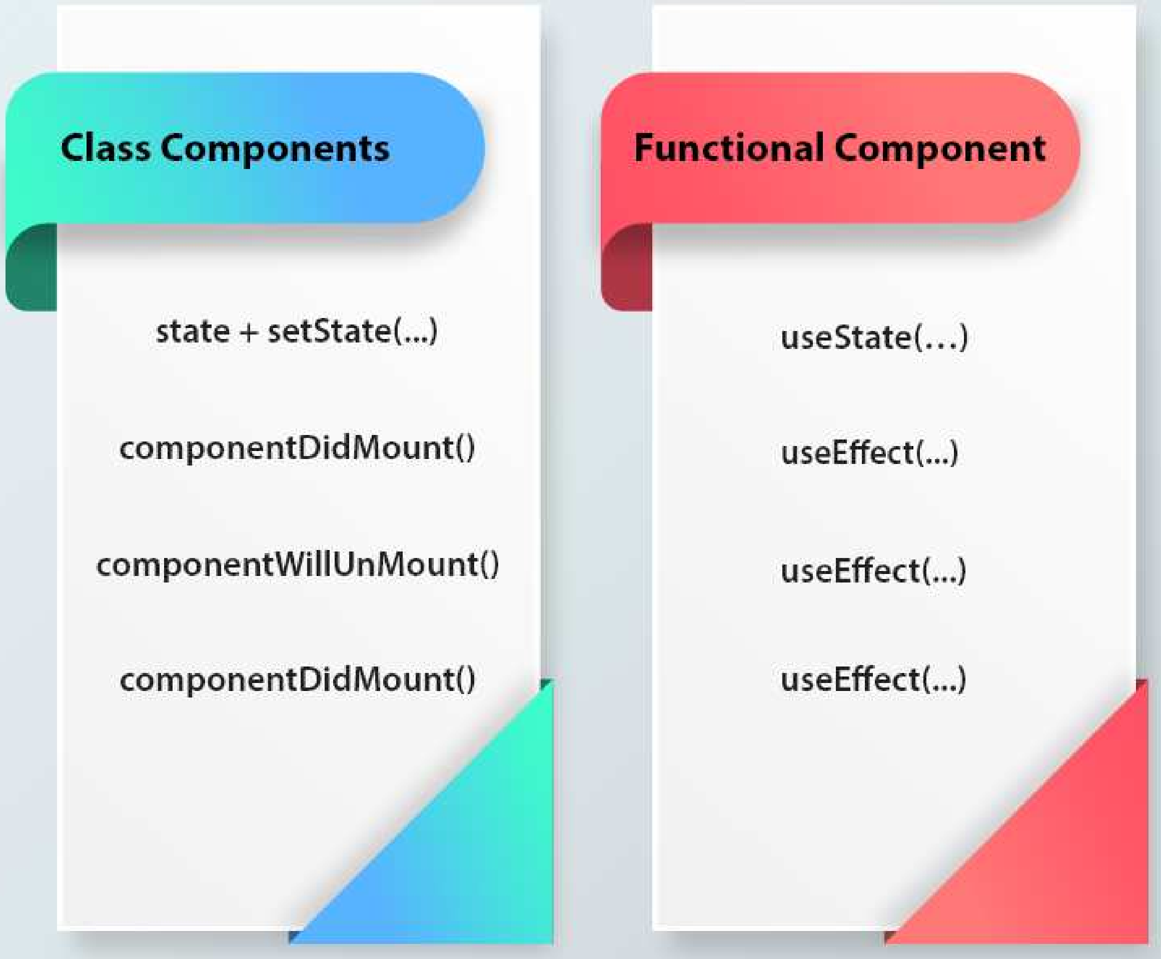
- State is a JavaScript Object that we use to track and response to users’ inputs
- Each React component has its own instance of state
- Most important thing about state:
  - **Any change in state** will cause all components or any children components inside of it to **be re- rendered**

https://reactnative.dev/docs/state



React Native Hooks

React Hooks



Class Components	Functional Component
state + setState(...)	useState(…)
componentDidMount()	useEffect(...)
componentWillUnMount()	useEffect(...)
componentDidMount()	useEffect(...)

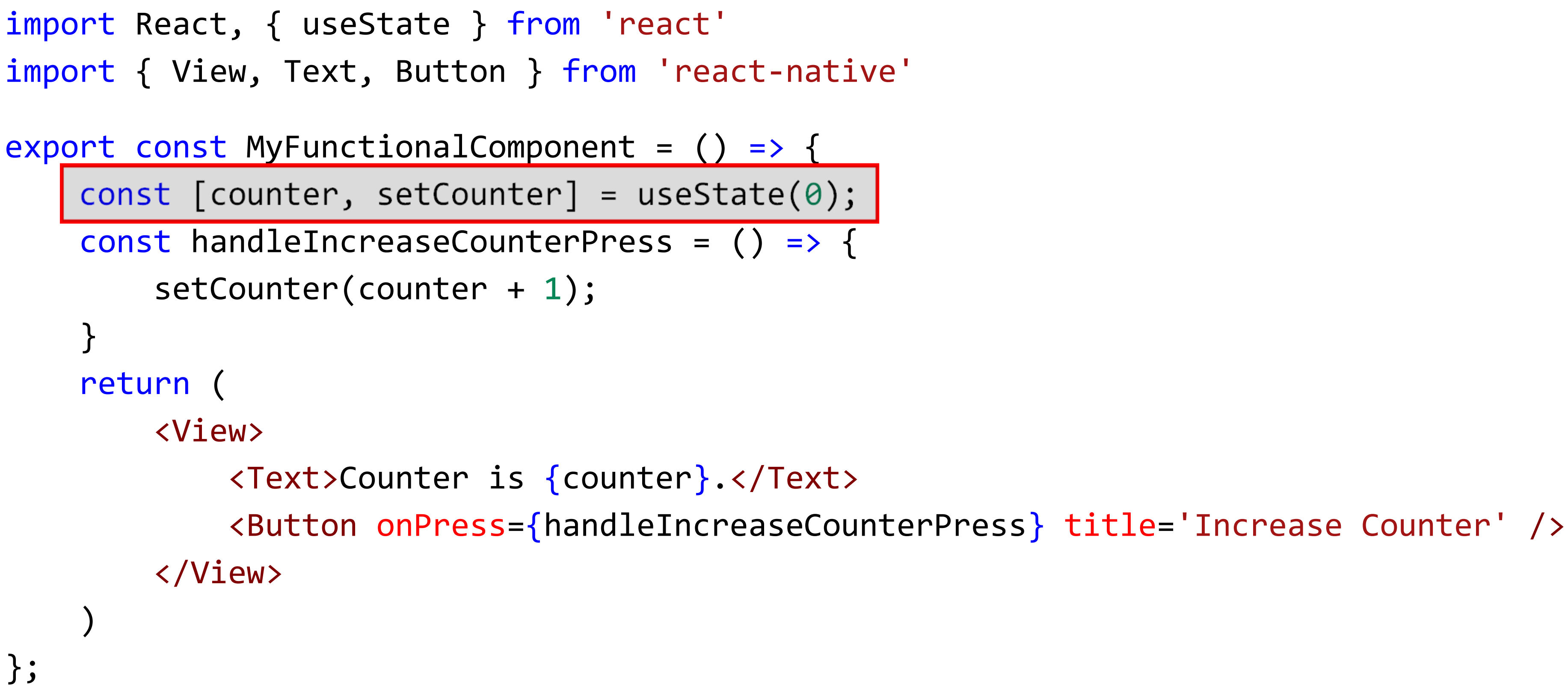
https://www.bacancytechnology.com/blog/react-native-hooks-to-build-app

What is React Hooks?

- Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class. It mainly uses to handle the state and side effects in react functional component. React Hooks are a way to use stateful functions inside a functional component.

- To manage state of React App, we have many hooks such as useState, useEffect, useContext, …
- Refs: https://github.com/react-native-community/hooks

useState



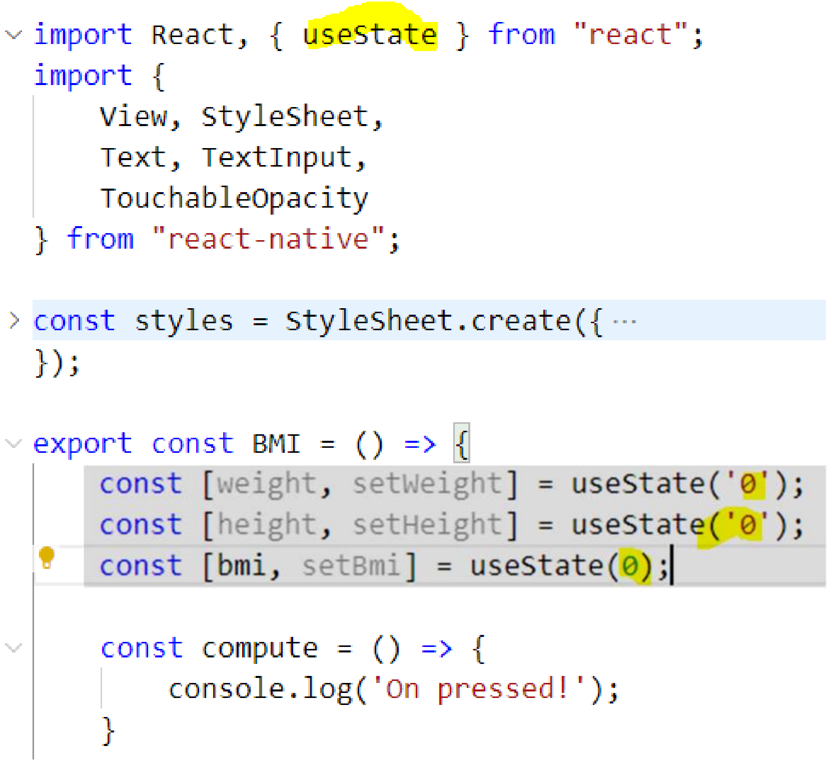
```
import React, { useState } from 'react'
import { View, Text, Button } from 'react-native'

export const MyFunctionalComponent = () => {
    const [counter, setCounter] = useState(0);
    const handleIncreaseCounterPress = () => {
        setCounter(counter + 1);
    }
    return (
        <View>
            <Text>Counter is {counter}.</Text>
            <Button onPress={handleIncreaseCounterPress} title='Increase Counter' />
        </View>
    )
};
```

const [state, setState] = useState(defaultValue);

HIENLTH

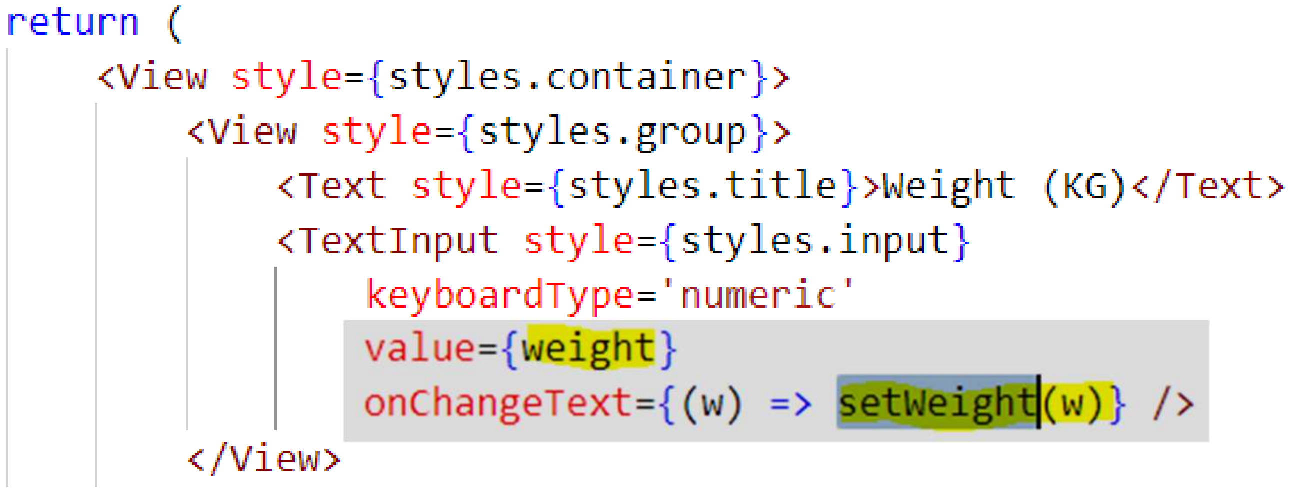
Initialize State

- We first defined 3 variables in state
  - **weight (String) = ‘0’** - It is a string because we use it with TextInput
  - **height (String) = ‘0’** - It is a string because we use it with TextInput
  - and calculated **bmi (Number) = 0**



```
import React, { useState } from "react";
import {
    View, StyleSheet,
    Text, TextInput,
    TouchableOpacity
} from "react-native";

const styles = StyleSheet.create({ ···
});

export const BMI = () => {
    const [weight, setWeight] = useState('0');
    const [height, setHeight] = useState('0');
    const [bmi, setBmi] = useState(0);

    const compute = () => {
        console.log('On pressed!');
    }
```

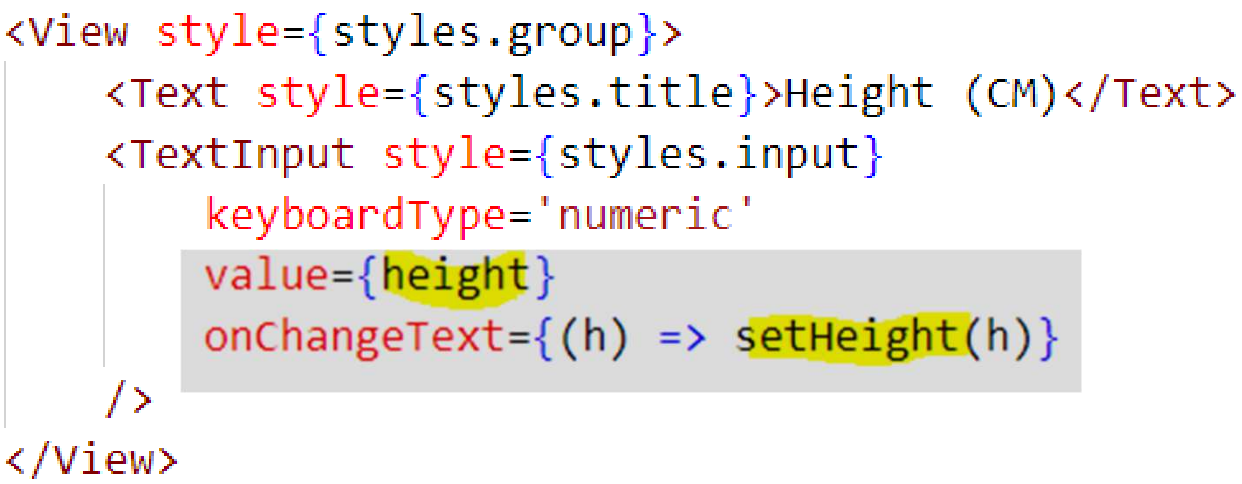
Showing/Setting State In TextInput



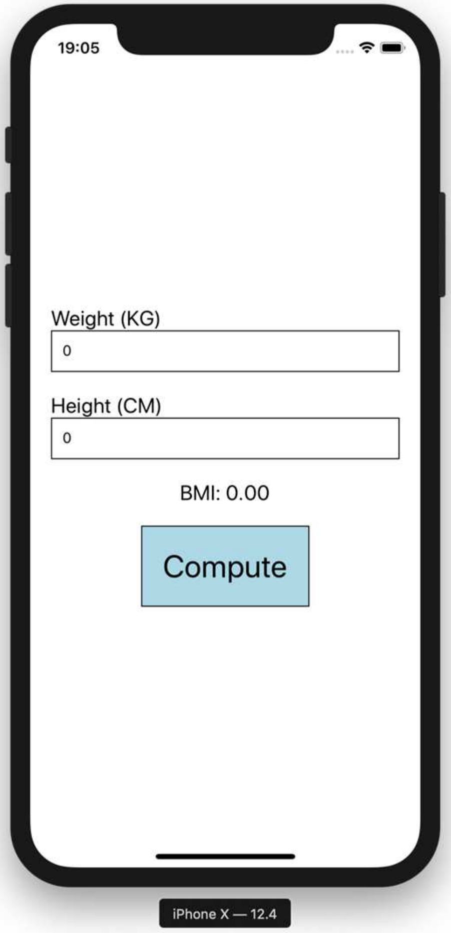
```
return (
    <View style={styles.container}>
        <View style={styles.group}>
            <Text style={styles.title}>Weight (KG)</Text>
            <TextInput style={styles.input}
                keyboardType='numeric'
                value={weight}
                onChangeText={(w) => setWeight(w)} />
        </View>
```

- **value attribute** - setting the text according to the state.weight’s value
- **onChangeText attribute** - invoked when user making change the TextInput, apply the new input value into the state

Continue On Height TextInput



```
<View style={styles.group}>
    <Text style={styles.title}>Height (CM)</Text>
    <TextInput style={styles.input}
        keyboardType='numeric'
        value={height}
        onChangeText={(h) => setHeight(h)}
    />
</View>
```



19:05

Weight (KG)

0

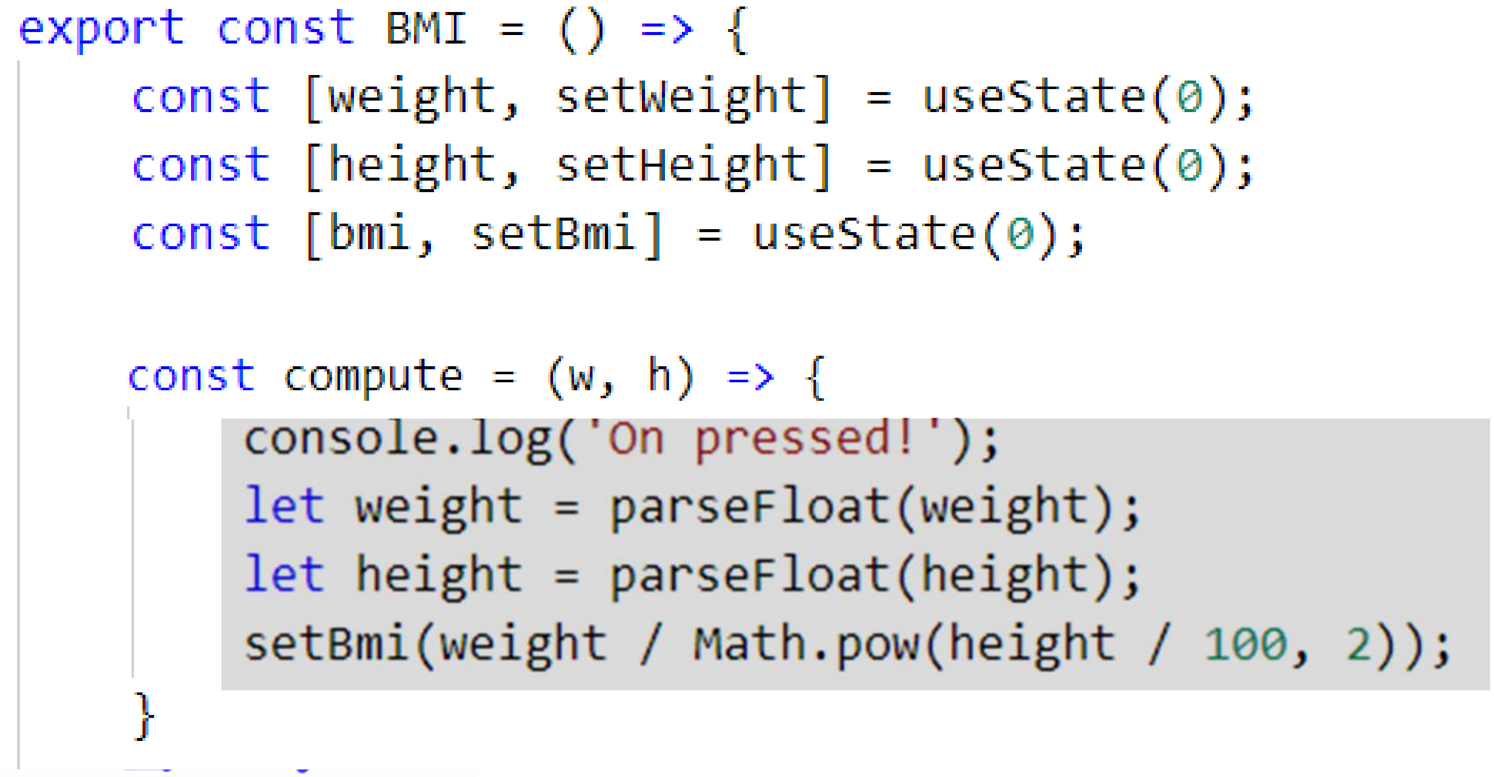
Height (CM)

0

BMI: 0.00

Let’s Compute The BMI

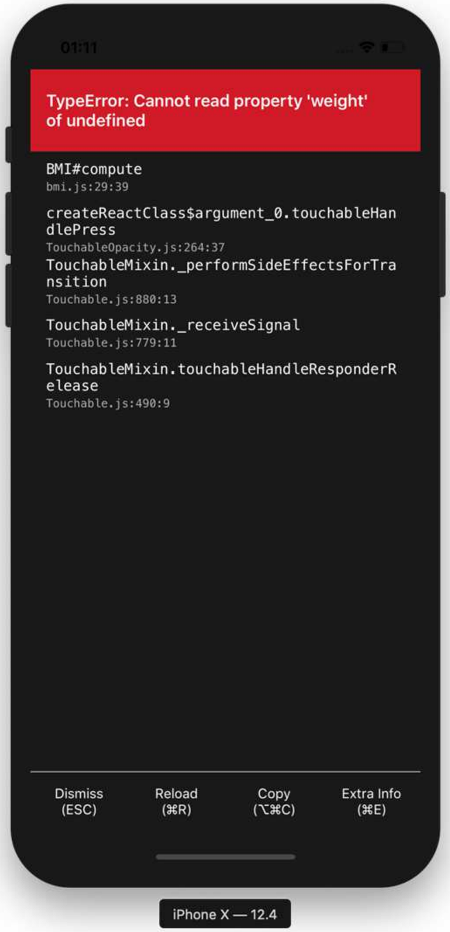
$$\mathrm{BMI} = \frac{\mathrm{mass(kg)}}{(\mathrm{height(m)})^2}$$



```
export const BMI = () => {
    const [weight, setWeight] = useState(0);
    const [height, setHeight] = useState(0);
    const [bmi, setBmi] = useState(0);

    const compute = (w, h) => {
        console.log('On pressed!');
        let weight = parseFloat(weight);
        let height = parseFloat(height);
        setBmi(weight / Math.pow(height / 100, 2));
    }
```

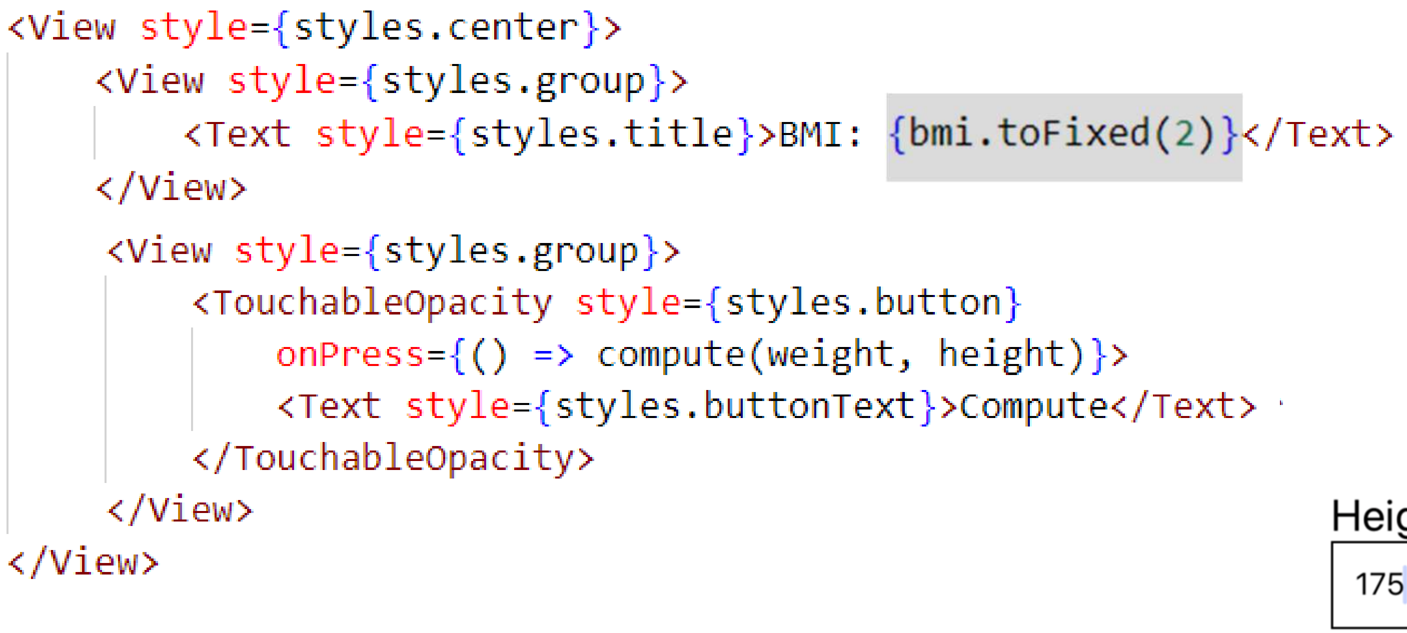
Method Biding

- Refresh the app now, input some number, and push compute button. You will see the red screen.
- It is because `compute()` method is not the standard method in the component class (We added it by our own)
- The scope of `compute()` method is not yet visible to the class’s state.

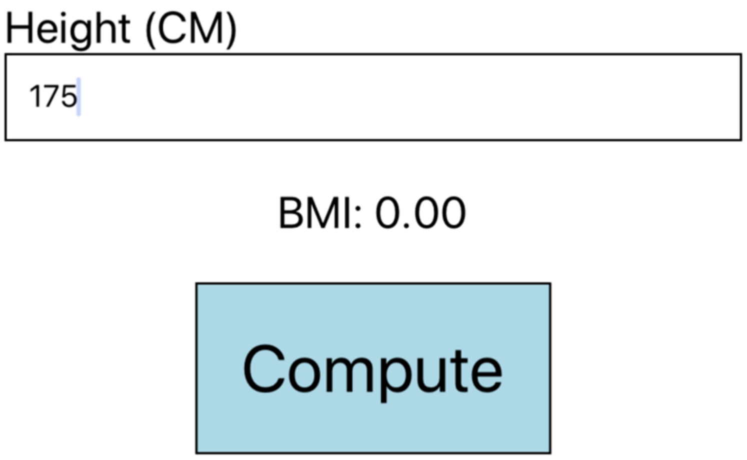


```
TypeError: Cannot read property 'weight'
of undefined

BMI#compute
bmi.js:29:39

createReactClass$argument_0.touchableHan
dlePress
TouchableOpacity.js:264:37
TouchableMixin._performSideEffectsForTra
nsition
Touchable.js:880:13

TouchableMixin._receiveSignal
Touchable.js:779:11

TouchableMixin.touchableHandleResponderR
elease
Touchable.js:490:9
```

Showing The Output



```
<View style={styles.center}>
    <View style={styles.group}>
        <Text style={styles.title}>BMI: {bmi.toFixed(2)}</Text>
    </View>
    <View style={styles.group}>
        <TouchableOpacity style={styles.button}
            onPress={() => compute(weight, height)}>
            <Text style={styles.buttonText}>Compute</Text>
        </TouchableOpacity>
    </View>
</View>
```

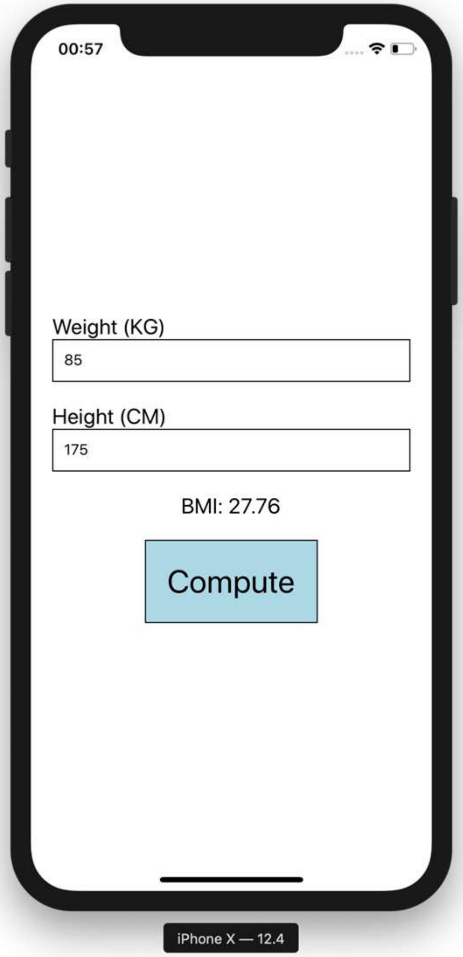


Height (CM)

175

BMI: 0.00

BMI Final



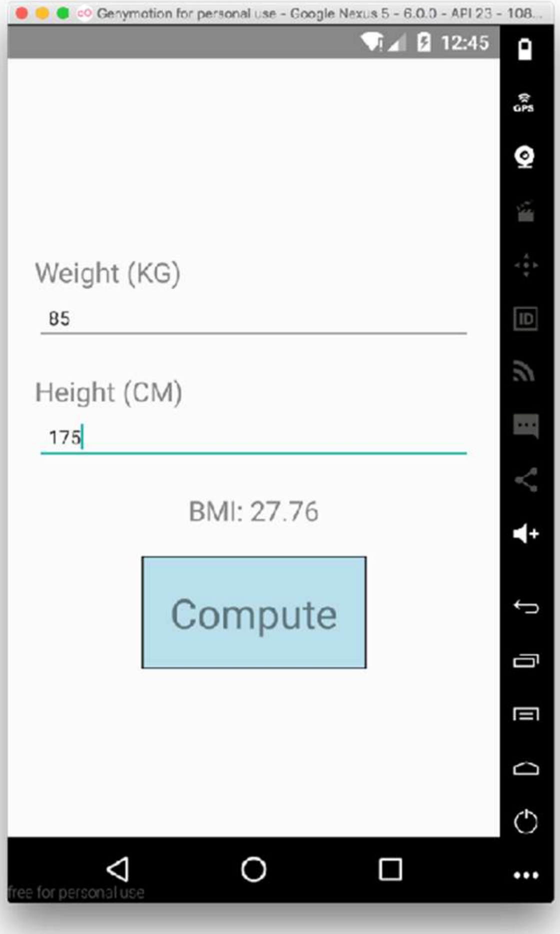
Weight (KG)

85

Height (CM)

175

BMI: 27.76


Genymotion for personal use - Google Nexus 5 - 6.0.0 - API 23 - 108...
12:45

Weight (KG)

85

Height (CM)

175

BMI: 27.76

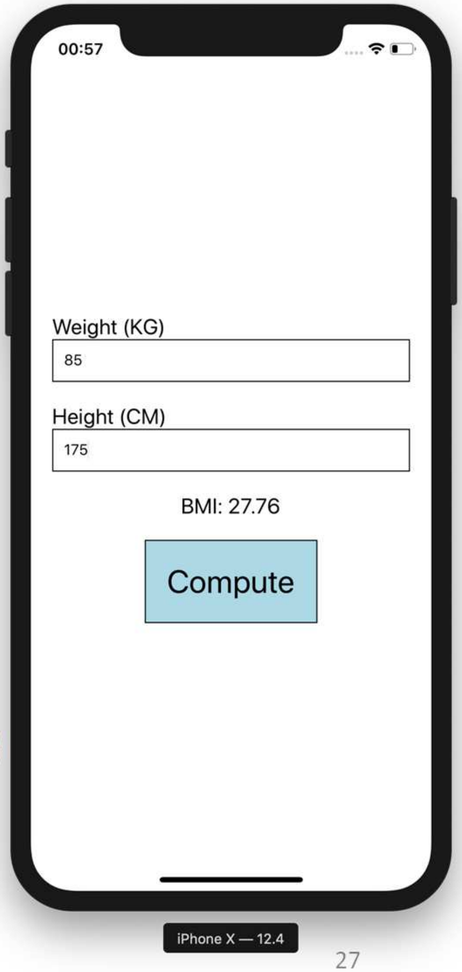
Compute
free for personal use

Exercise 1

- Showing the obesity level under the BMI number.
  - BMI > 32 = Obese
  - 25 < BMI < 32 = Over Weight
  - 18.5 < BMI < 25 = Normal Weight
  - BMI < 18.5 = Under Weight

- Declare state to store all BMI score

- Add button ‘Clear’
  Clear all BMI calculation (remove state list of BMI Store)



Weight (KG)
85

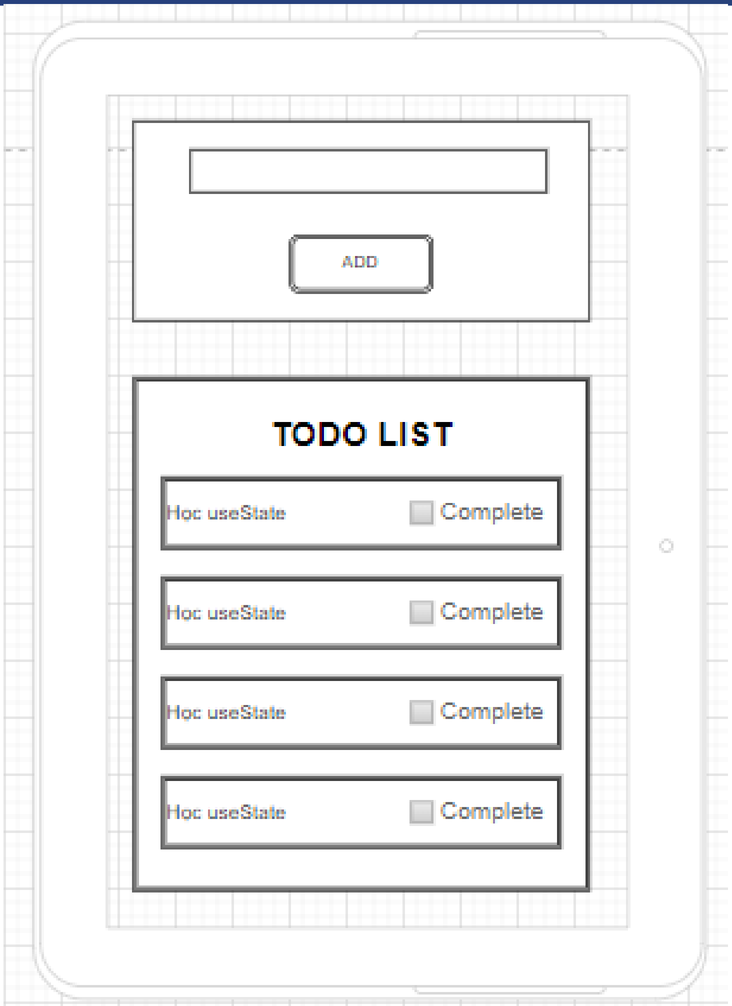
Height (CM)
175

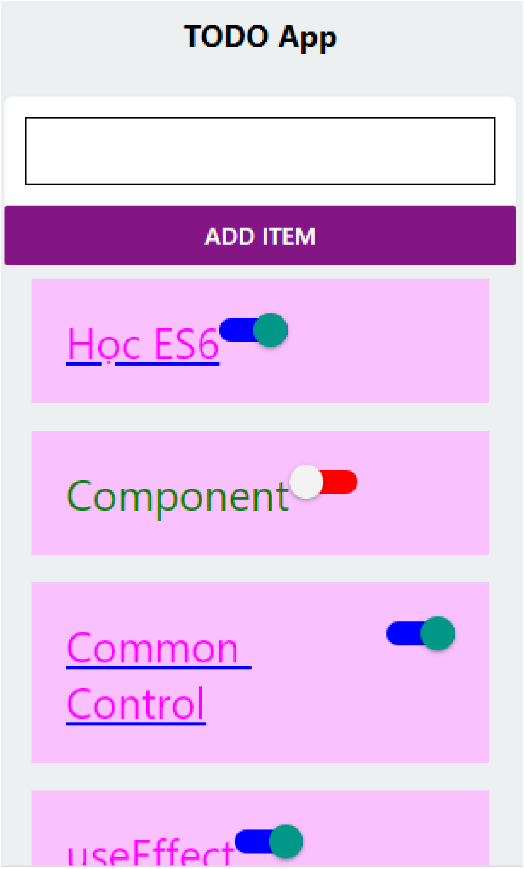
BMI: 27.76

Exercise 2: Todo app

Design an to do app including:

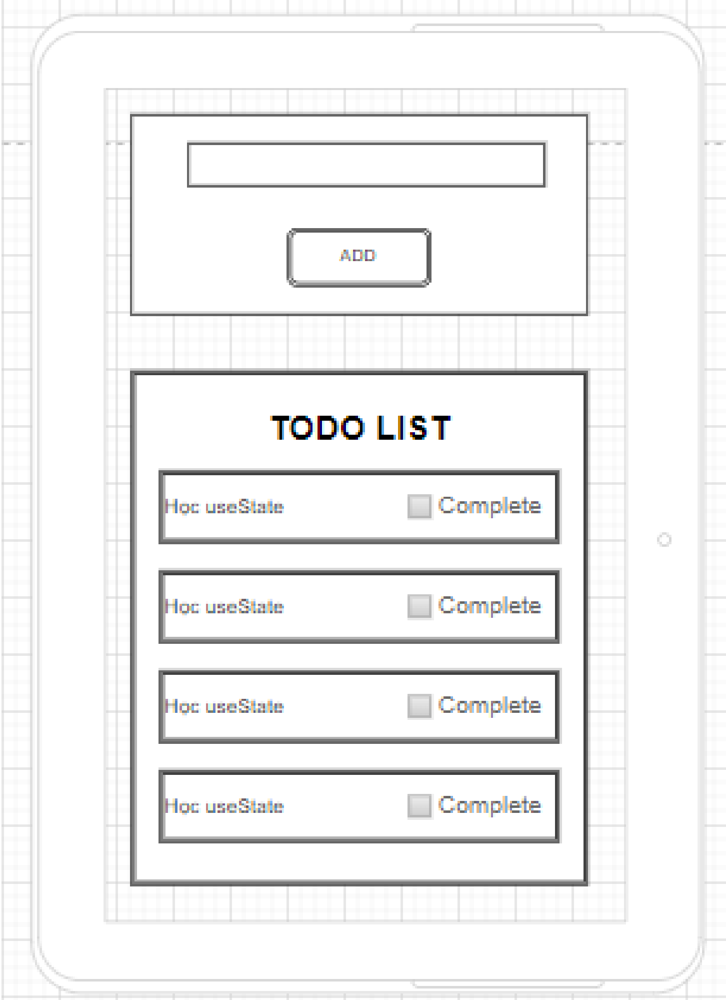
- Add task
- Show tasks. Hint:
  - FlatList, Switch
  - https://docs.expo.dev/versions/latest/react-native/flatlist/
- For each task in list, user can mark DONE

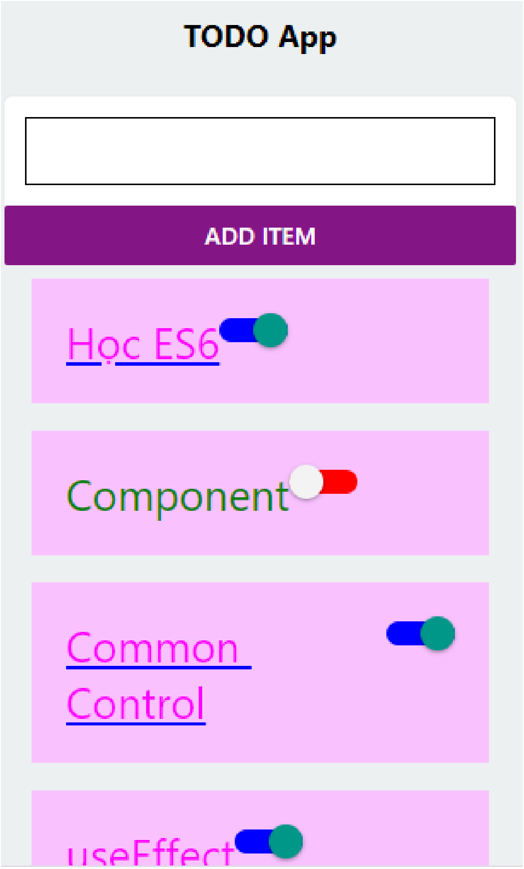




Exercise 2: TODO App Advanced

- Adding search/filter area base on task name





Thank you!