Activity No. 3	
Sampling Live Data	
Course Code: CPE026	**Program:** CPE
Course Title: Emerging Technologies 3	**Date Performed:** 9/28/2023
Section: CPE41S3	**Date Submitted:** 9/28/2023
Name/s: Aducal, John Mark Alcaraz, Arwen Christine Castillo, John Lloyd Renzo Sanchez, Krizelle	**Instructor:** Engr. Roman M. Richard

1. Instructions:

- Perform the given tasks.
- Follow the Hands-on Activity format from the previous activities to accomplish the given instructions below.
- Perform the Procedures and Outputs in Section 6.
- Answer the Supplementary Activity in Section 7.
- Provide the following in section 8:
- A concise summary of the activity performed.
- Your personal conclusions / reflections.
- An overview of the lessons you learned from this module.
- Submission Requirements:
- All files must be submitted as PDF only.

2. Tasks:

1. Go through the topics presented in the module: Determine the difference between a button and a pressable?

- The difference between a button and a pressable component in React Native lies in their flexibility and customization. To compare the two, upon using both components we were able to see that the button component provides a straightforward button element, but it has limited customization options. On the other hand, the pressable component provides a greater control over touch interactions.

2. Create a pressable component in your goal list application instead of a button for 'Add Goal'. How is this any different?

- Using a pressable component for ‘Add Goal’ combines the text insertion with other interactive elements. What makes it different from a button is its straightforwardness and does not react to touch gestures like long-press or double-tap.

3. Experiment with the different props of the pressable component.

- In our code, we used different props such as ‘onPress’, ‘onLongPress’, ‘android_ripple’ , and ‘hitSlop’.

Additional Screenshots:
Arwen's pressable and layout design

```
<Pressable
  style={({ pressed }) => [

    pressed && styles.pressedButton,
  ]}
  onPress={addGoalHandler} // Change onPress to onLongPress
  android_ripple={{ color: 'rgba(0, 0, 0, 0.1)' }}
  hitSlop={{ top: 10, bottom: 10, left: 10, right: 10 }}
>
  <Text style={styles.addButtonLabel}>Add Goal</Text>
</Pressable>
```

Figure 1. Code to apply pressable component

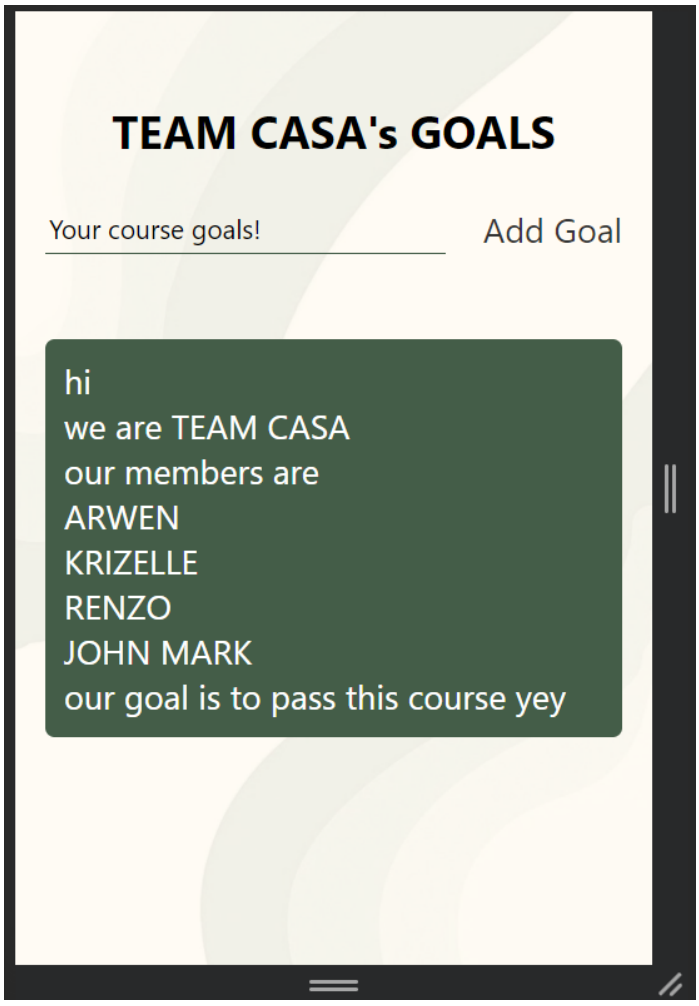


Figure 2. Output after deploying to web

Renzo's pressable and layout design

```
<Pressable
  style={({ pressed }) => [

    pressed && styles.pressedButton,
  ]}
  onPress={addGoalHandler}
  android_ripple={{ color: 'rgba(0, 0, 0, 0.1)' }}
  hitSlop={{ top: 10, bottom: 10, left: 10, right: 10 }}
>
  <Text style={styles.addButtonLabel}>Add Goal</Text>
</Pressable>
```

Figure 3. Code to apply pressable component

Figure 4. Output after deploying to iOS using ExpoGo

Krizelle's pressable and layout design

```
/>
<Pressable
  style={({ pressed }) => [

    pressed && styles.pressedButton,
  ]}
  onPress={addGoalHandler}
  android_ripple={{ color: 'green' }}
>
  <Text style={styles.addButtonLabel}>Add Goal</Text>
</Pressable>
```

Figure 5. Code to apply pressable component

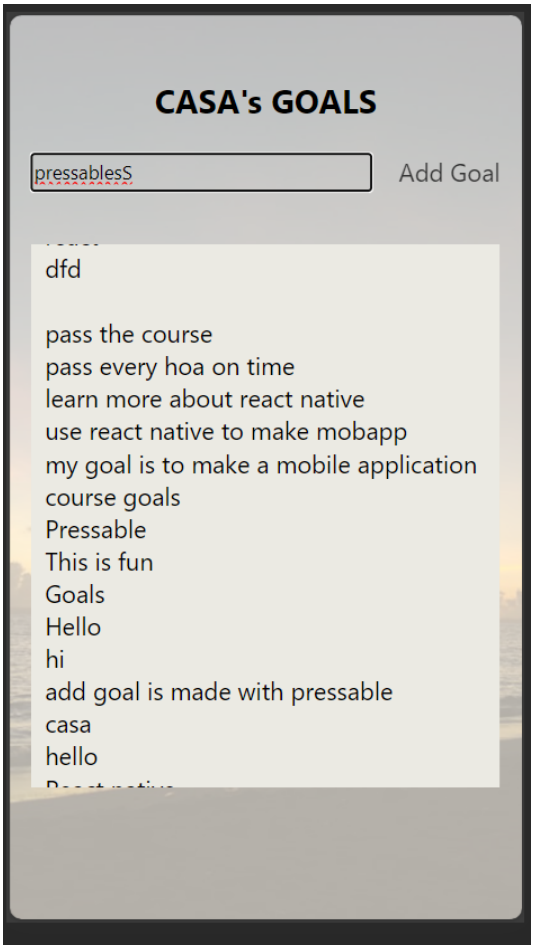
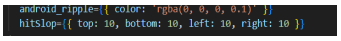


Figure 6. Output after deploying to web

3. Supplementary Activity:

1. Create a table and show the different props of the pressable component.
2. Create a column for the description of each prop.
3. Create a column that shows the syntax of each prop.
4. Create a column that shows your demonstration of each prop's use.

PROP	DESCRIPTION	SYNTAX	DEMONSTRATION
onPress	Used when pressable is pressed	onPress={() => { /* Your function here */ }}	
onLongPress	Used when pressable is long pressed	onLongPress={() => { /* Your function here */ }}	
android-ripple	Ripple effect on android	{ color: 'rgba(0, 0, 0, 0.1)' }	
hitSlop	Extends touchable area around the pressable	hitSlop={{ top: 10, bottom: 10, left: 10, right: 10 }}	

4. Conclusions:

Summary of Activity Performed:
In the activity focused on using the "Pressable" component in React Native, the team was able to gain valuable insights about how interactive and responsive user interfaces can be done using the said component. Upon doing the activity, we were able to discover that "Pressable" is a versatile tool to use when creating various press interactions, such as taps and long presses. By learning its properties like "onPress" and "onLongPress," we could activate the component and apply actions based on user input.

Personal conclusions / reflections:

Aducal - In this activity, I learned the difference between a button and a pressable. while the button is pre-styled with limited customization options. The pressable has greater flexibility and touch interactions ideally for complex UI components.

Alcaraz - After doing the activity about the utilization of the "Pressable" component in React Native, I've gained a deeper understanding of its purpose and its effect on enhancing user inputs within mobile

applications. Learning the use of pressable allowed me to create an engaging and user-friendly interface which I could say that I can apply in our final project.

Castillo- This activity teaches me the uses of the “Pressable” function. After applying it to our class activities, I saw that the Pressable is way better as it promotes cleanliness and is much more lively to use than the “TouchableOpacity”.

Sanchez- In this activity, I have learned to use Pressables. Employing this builds an engaging and aesthetic UI for the application. Utilizing the different props, we have enhanced the component ‘Add Goal’ from being a button to a pressable which makes it more interactive and visually pleasing.