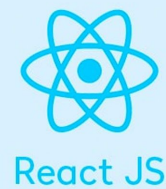


Structuring React Native Code— Wireless Buzzer App



What we did:

- Used Stylesheet to better name and use CSS styles.
- Exported and imported custom React Native components.
- Wrote structured code for each custom React Native component differently.

How we did it:

In this class, our task was to create a screen which would allow users to pick their team and then navigate to the buzzer button. We used a sheet to style the code as we do in HTML and CSS.



In react native, a good practice is to create a separate styles object using 'StyleSheet.create()' and use the declared styles inside the style prop of the component when rendering.

- First, we import the 'StyleSheet' component defined in React Native.
- Second, we use 'StyleSheet.create()' to create the different styles. The function expects a JSON object with different styles defined on them.
- Note: We use "const" to create a variable which we do not want to change while the program is running. Typically we do not want the style object to change during the program.
- We can call the styles.<name of the style> inside the style prop of a component to give a particular style to the React Native component.

```
1  import * as React from 'react';
2  import { Text, View, Button, TouchableOpacity, StyleSheet } from 'react-native';
3  import { Audio } from 'expo-av';
4
5  class SoundButton extends React.Component {
6    playSound = async () => {
7      await Audio.Sound.createAsync(
8        { uri: 'http://soundbible.com/mp3/Buzzer-SoundBible.com-188422102.mp3' },
9        { shouldPlay: true }
10     );
11   }
12
13   render() {
14     return (
15       <TouchableOpacity
16         style={{
17           marginLeft: 100,
18           borderWidth: 1,
19           borderColor: 'rgba(0,0,0,0.2)',
20           alignItems: 'center',
21           justifyContent: 'center',
22           width: 200,
23           height: 200,
24           backgroundColor: 'red',
25           borderRadius: 100,
26         }}
27         onPress={this.playSound}>
28         <Text
29           style={{
```

Prettier {} Editor ⚙️ ⓘ

```

32 export default class App extends React.Component {
33   render() {
34     return (
35       <View>
36         <SoundButton />
37       </View>
38     );
39   }
40 }
41
42 const styles = StyleSheet.create({
43   button: {
44     marginTop: 200,
45     marginLeft: 100,
46     borderWidth: 1,
47     borderColor: 'rgba(0,0,0,0.2)',
48     alignItems: 'center',
49     justifyContent: 'center',
50     width: 200,
51     height: 200,
52     backgroundColor: 'red',
53     borderRadius: 100,
54   },
55   buttonText: {
56     fontWeight: 'bold',
57     fontSize: 20,
58   }
59 });

```

ios

Press Me

```

4
5 class SoundButton extends React.Component {
6   playSound = async () => {
7     await Audio.Sound.createAsync(
8       { url: 'http://soundbible.com/mp3/Buzzer-SoundBible.com-188422102.mp3' },
9       { shouldPlay: true }
10    );
11  }
12
13  render() {
14    return (
15      <TouchableOpacity
16        style={styles.button}
17        onPress={this.playSound}>
18        <Text
19          style={styles.buttonText}>
20          Press Me
21        </Text>
22      </TouchableOpacity>
23    );
24  }
25 }
26
27 export default class App extends React.Component {
28   render() {
29     return (
30       <View>
31         <SoundButton />
32       </View>

```

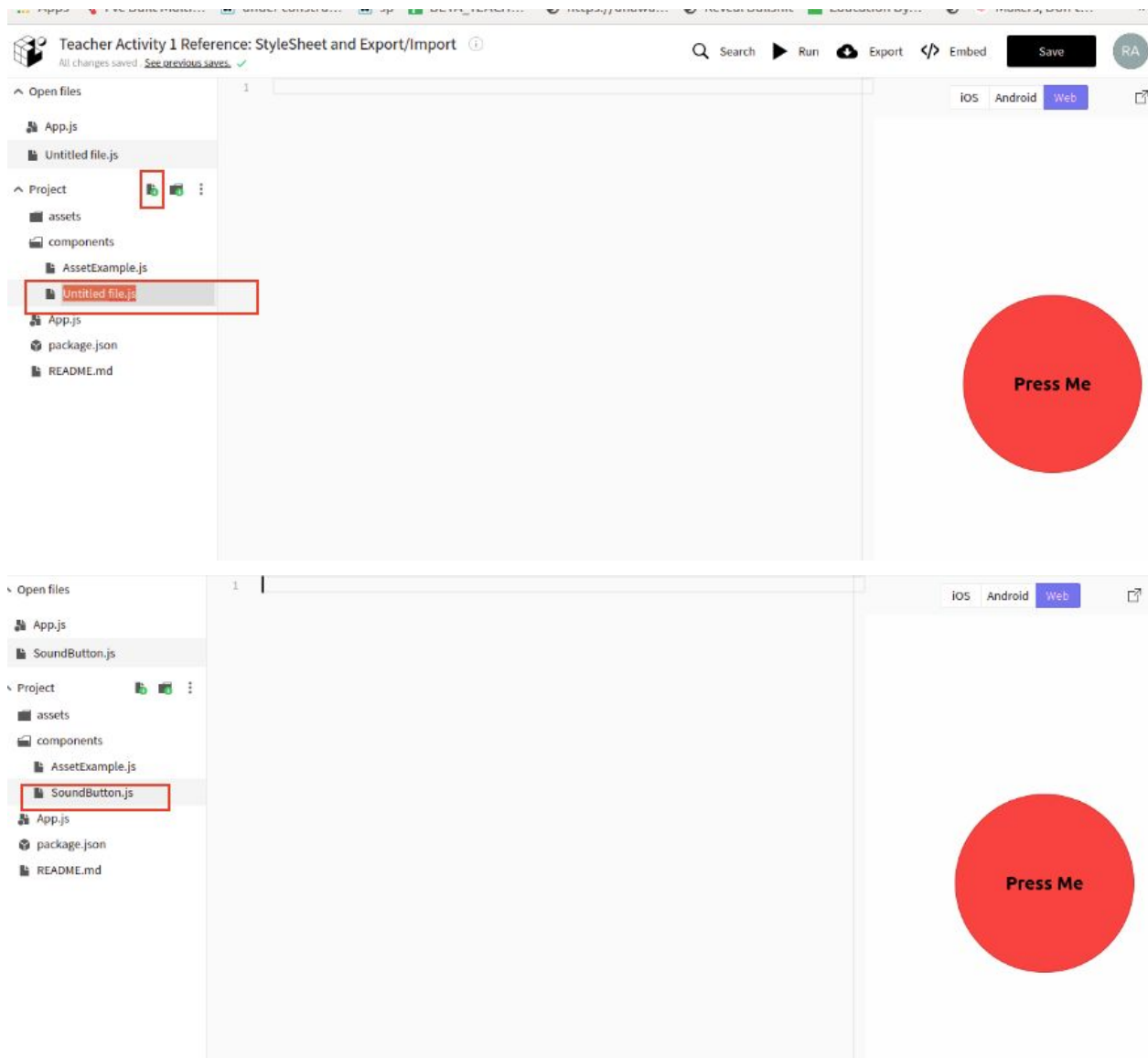
ios Android Web

Press Me

Prettier {} Editor Expo v36.0.0 Devices 1 Preview

We created different components in different files to keep things more organized and then import all the components in App.js to use them.

- Create a new file inside the components folder and call it 'SoundButton.js' (since it defines the SoundButton Component).
- Copy all the code for the 'SoundButton' component into this file.
- Import components from the libraries used for this component including - react, expo-av , StyleSheet etc.
- Add an export statement - export default SoundButton.
- This instruction allows the SoundButton component to be imported by default when we write 'import' statement in another file.





Teacher Activity 1 Reference: StyleSheet and Export/Import ⓘ

All changes saved 2 minutes ago. [See previous saves.](#) ✓

Search



Run



Open files

App.js

SoundButton.js

Project

assets

components

AssetExample.js

SoundButton.js

App.js

package.json

README.md

```

1  import * as React from 'react';
2  import { Text, View, Button, TouchableOpacity, StyleSheet } from 'react-native';
3  import { Audio } from 'expo-av';
4
5  class SoundButton extends React.Component {
6    playSound = async () => {
7      await Audio.Sound.createAsync(
8        { uri: 'http://soundbible.com/mp3/Buzzer-SoundBible.com-188422102.mp3' },
9        { shouldPlay: true }
10     );
11   }
12
13   render() {
14     return (
15       <TouchableOpacity
16         style={styles.button}
17         onPress={this.playSound}>
18         <Text
19           style={styles.buttonText}>
20           Press Me
21         </Text>
22       </TouchableOpacity>
23     );
24   }
25 }
26
27 export default class App extends React.Component {
28   render() {
29     return (

```

✓ No errors

Prettier {} Editor



Teacher Activity 1 Reference: StyleSheet and Export/Import ⓘ

All changes saved. [See previous saves.](#) ✓

Search



Run



Open files

App.js

AssetExample.js

SoundButton.js

Project

assets

components

AssetExample.js

SoundButton.js

App.js

package.json

README.md

```

1  import * as React from 'react';
2  import { Text, View, TouchableOpacity, StyleSheet } from 'react-native';
3  import { Audio } from 'expo-av';
4
5  class SoundButton extends React.Component {
6    playSound = async () => {
7      await Audio.Sound.createAsync(
8        { uri: 'http://soundbible.com/mp3/Buzzer-SoundBible.com-188422102.mp3' },
9        { shouldPlay: true }
10     );
11   }
12
13   render() {
14     return (
15       <TouchableOpacity
16         style={styles.button}
17         onPress={this.playSound}>
18         <Text
19           style={styles.buttonText}>
20           Press Me
21         </Text>
22       </TouchableOpacity>
23     );
24   }
25 }
26
27 export default SoundButton;

```

Teacher Activity 1 Reference: StyleSheet and Export/Import

Saving changes... See previous saves.

Open files

- App.js
- AssetExample.js
- SoundButton.js

Project

- assets
- components
 - AssetExample.js
 - SoundButton.js
- App.js
- package.json
- README.md

```

17     onPress={this.playSound}>
18     <Text
19       style={styles.buttonText}>
20       Press Me
21     </Text>
22   </TouchableOpacity>
23   };
24 }
25 }
26
27 const styles = StyleSheet.create({
28   button: {
29     marginTop: 200,
30     marginLeft: 100,
31     borderWidth: 1,
32     borderColor: 'rgba(0,0,0,0.2)',
33     alignItems: 'center',
34     justifyContent: 'center',
35     width: 200,
36     height: 200,
37     backgroundColor: 'red',
38     borderRadius: 100,
39   },
40   buttonText: {
41     fontWeight: 'bold',
42     fontSize: 20,
43   }
44 });
45
46 export default SoundButton;

```

✓ No errors

Prettier {} Editor ⚙️ Exp

Import SoundButton from the components folder.

Teacher Activity 1 Reference: StyleSheet and Export/Import

Last saved less than 5 seconds ago. See previous saves.

Open files

- App.js
- AssetExample.js
- SoundButton.js

Project

- assets
- components
 - AssetExample.js
 - SoundButton.js
- App.js
- package.json
- README.md

```

1 import * as React from 'react';
2 import { View } from 'react-native';
3
4
5
6
7 export default class App extends React.Component {
8   render() {
9     return (
10      <View>
11        <SoundButton />
12      </View>
13    );
14  }
15 }
16
17

```

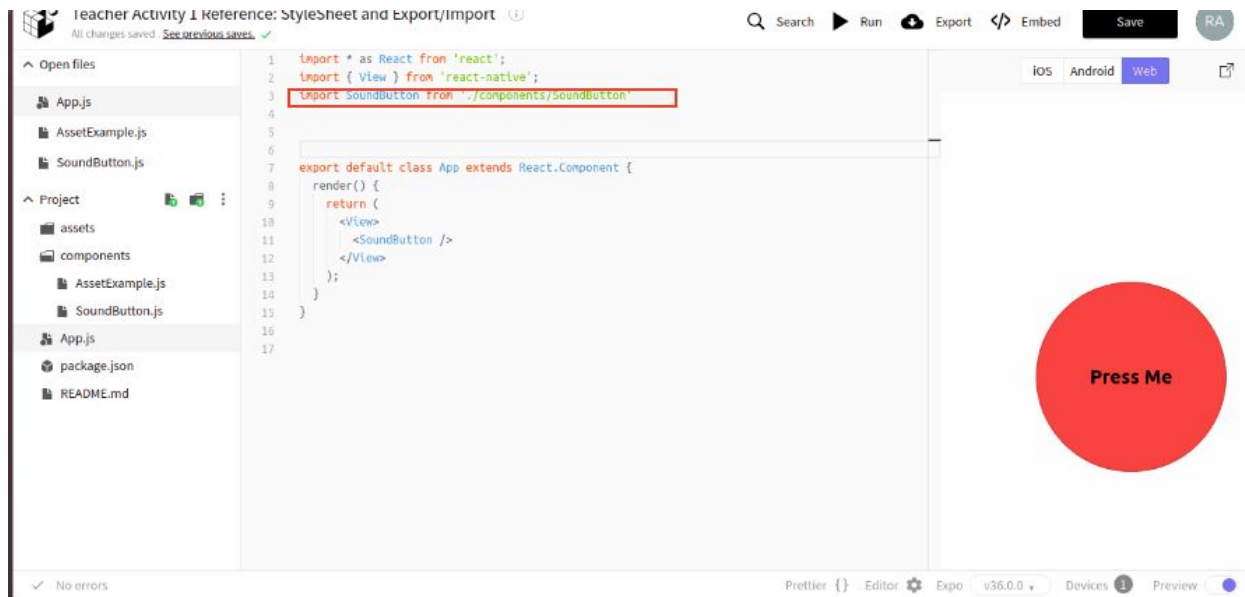
ios Android Web

Did you know:
You can turn off automatic updates under Devices in the footer?

SoundButton is not defined

ReferenceError: SoundButton is not defined

at App.render (module://App.js:jsitranspiled:40:79)
 at ga ([snack internals])
 at ha ([snack internals])
 at js ([snack internals])
 at js ([snack internals])
 at Ps ([snack internals])
 at ks ([snack internals])
 at [snack internals]
 at unstable_runWithPriority ([snack internals])
 at ui (https://s3-us-west-1.amazonaws.com/snack-web-player/36/static/js/2.d51b9d02.chunk.js:1:1046829)



Create a component called AppHeader which gives a header to your app (with app name):

Write a component 'AppHeader.js'.

Import necessary components at the top.

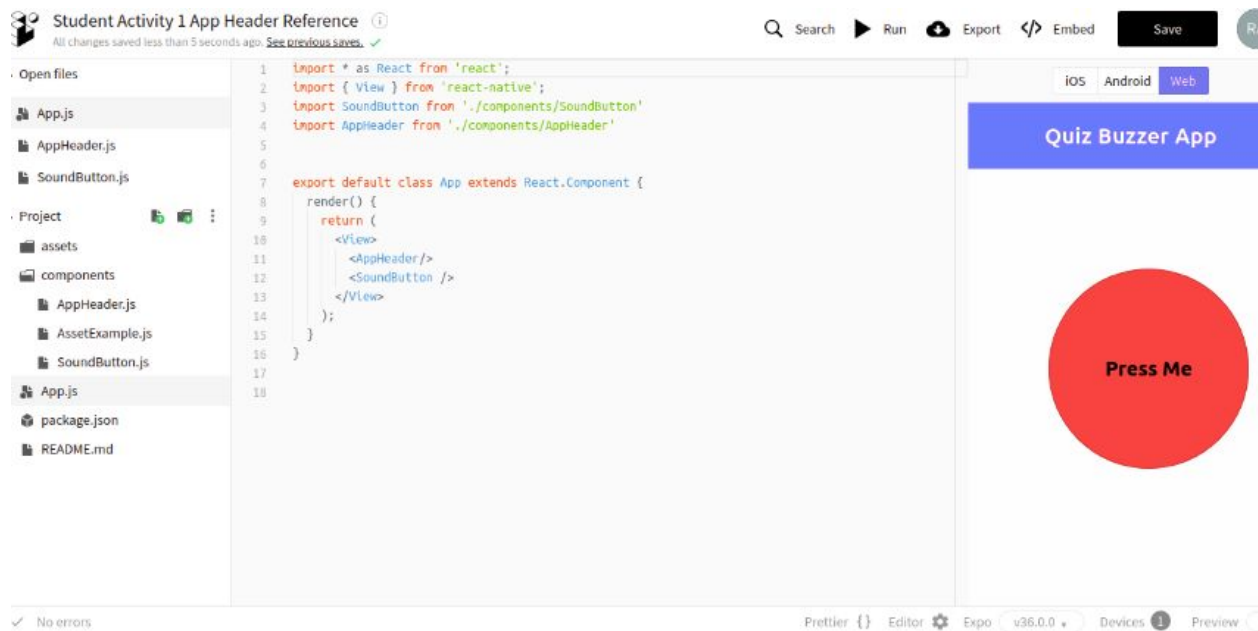
Render the component inside the render() function.

Create appropriate styling for it using StyleSheet

Export the class as the default export.



Use the 'AppHeader' component you created inside 'App.js' file.



What's next?:

In the next class, we will continue to work on the rest of the assigned tasks towards completing the Wireless Buzzer App. We will learn how to convert our single screen app to a two-screen app and how to navigate and pass information from one screen to another.