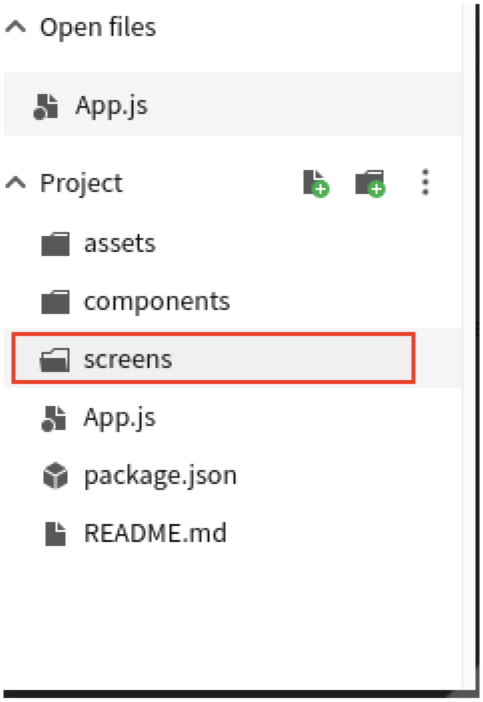
Topic	**React Navigation - Switch navigator**	
Class Description	**Students learn how to create a two-screen mobile app using Switch Navigator in React Native. They also learn to pass data from one screen to another.**	
Class	**C57**	
Class time	**45 mins**	
Goal	● Create Switch Navigation to move from one screen to another. ● Pass data from one screen to another.	
Resources Required	● Teacher Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Android/iOS Smartphone with Expo App installed ○ Expo Snack Account ● Student Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Android/iOS Smartphone with Expo App installed ○ Expo Snack Account	
Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 min** **5 min**

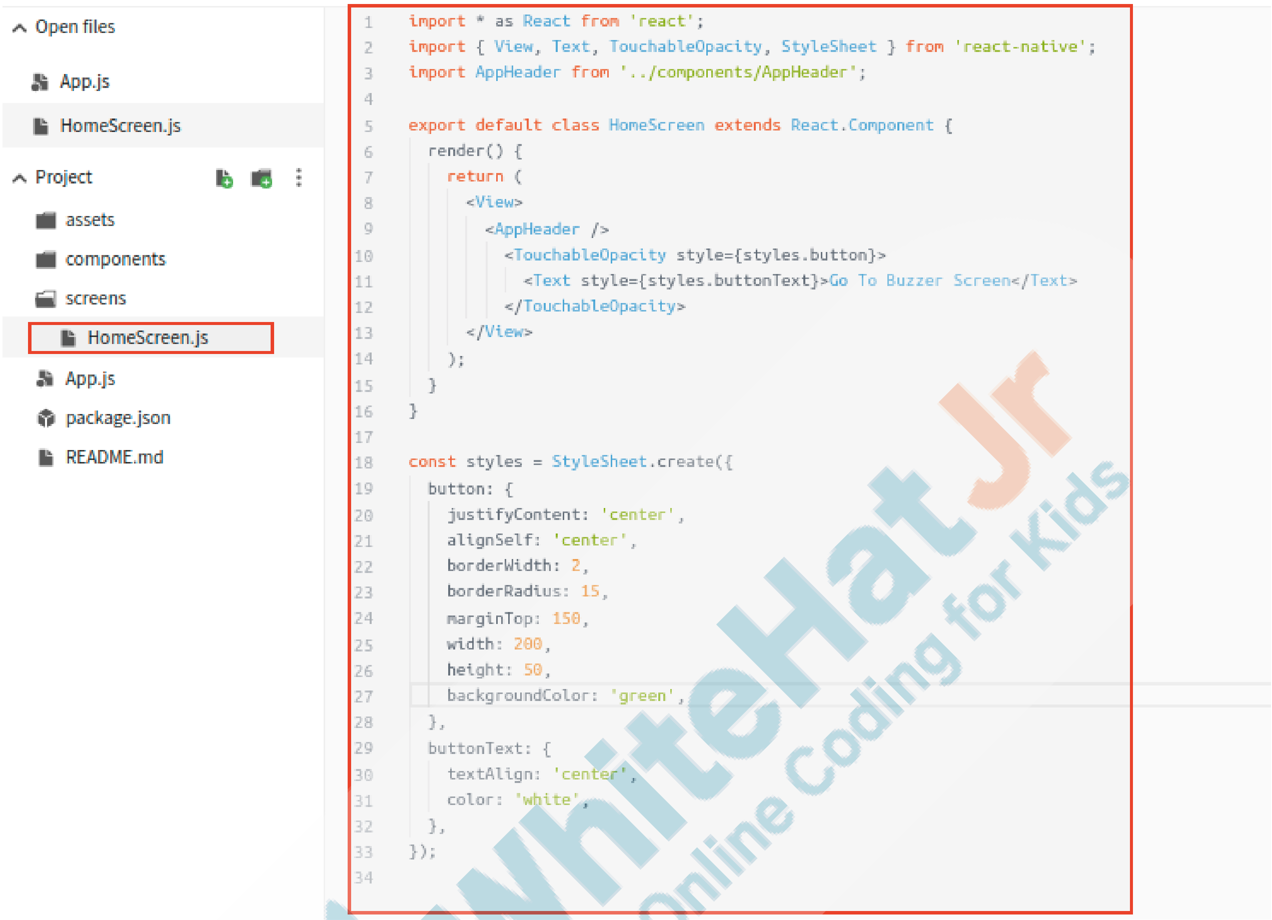
CONTEXT

- **Review code from the previous class.**
- **Introduce the problem of navigating from one screen to another.**

Class Steps	Teacher Action	Student Action
Step 1: **Warm Up** **(5 mins)**	Hey! Remember our plan for today's class?	ESR: We will learn to create a two screen app.
	Correct! But before we start, can we quickly review what we did in the last class?	ESR: - We learned to create style for the components using StyleSheet. - We learned to create independent components such as App Header, Buzzer Button, and how to export/import them in our files.
	Great! I am glad that you remember. In today's class, you will learn one of the ways to create a multi-screen app. We will create a button/switch. Clicking this button/switch will take us to another screen. This is called Switch Navigation in React Native. There are other Navigation techniques as well, which we will be exploring in the upcoming classes.	Student listens and asks questions.
	Alright, let's get hands on and get started!	-
Teacher Initiates Screen Share		

CHALLENGE ● **Create two screens as separate files.** ● **Create an App container which contains the two screens in the app.** ● **Create a switch navigator which switches from home screen to buzzer screen.**		
Step 2: Teacher-led Activity (15 min)	Teacher opens **Teacher Activity 1**. Before we make any changes in the code, let's quickly capture our current code and what we are doing here. Can you tell us what we have done in our code so far?	The student explains the progress in the code so far.
	You saw how we can create separate components as separate files in our React Native Code. We can also create two screens as separate files and then call them in our app. Let's create a "screens" folder in our file directory. This will hold all the different screens in our app as separate files. Teacher creates a folder called "screens"	The student observes and learns.

 Open files App.js Project assets components screens App.js package.json README.md		
	Let's create a screen/file in the folder called "HomeScreen.js" This file will contain a Button. We will use TouchableOpacity to do that. Pressing this button will take us to the next screen which will be our BuzzerScreen. Can you help me in creating this screen? Teacher takes student inputs to create "HomeScreen.js" containing a Button (TouchableOpacity). Note: Teacher can show the output under the Web Emulator on snack.	The student gives inputs on how to create a ‘TouchableOpacity’ Button on this screen. He/She also gives input on styling the button.

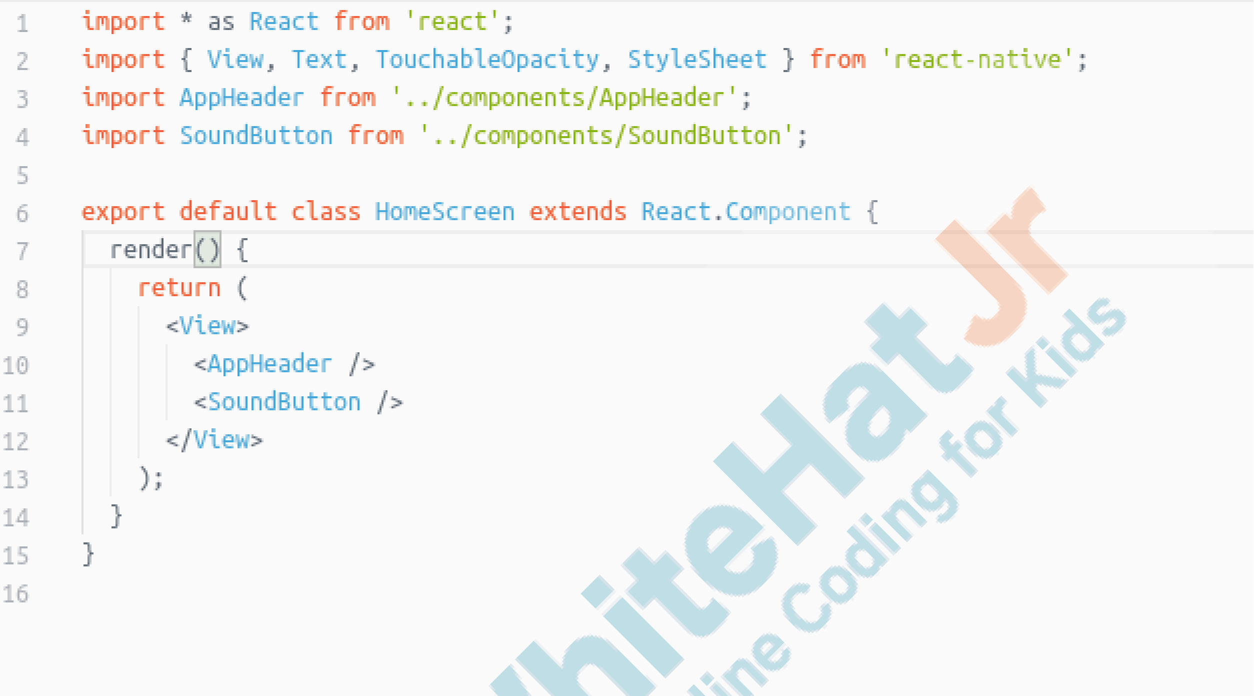


```
import * as React from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader';

export default class HomeScreen extends React.Component {
  render() {
    return (
      <View>
        <AppHeader />
          <TouchableOpacity style={styles.button}>
            <Text style={styles.buttonText}>Go To Buzzer Screen</Text>
          </TouchableOpacity>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  button: {
    justifyContent: 'center',
    alignSelf: 'center',
    borderWidth: 2,
    borderRadius: 15,
    marginTop: 150,
    width: 200,
    height: 50,
    backgroundColor: 'green',
  },
  buttonText: {
    textAlign: 'center',
    color: 'white',
  },
});
```

	Pressing the button should take us to the BuzzerScreen. Where should we write this code?	Inside onPress prop for TouchableOpacity.
	Correct. But we do not have a BuzzerScreen ready. So, before we do that, let's create a 'BuzzerScreen.js' file in our screens folder. We need to add our 'SoundButton' component to this screen. Can you help me create the BuzzerScreen?	The student guides to create 'BuzzerScreen' by adding the 'AppHeader' and 'SoundButton' Components in the file.

	Teacher creates ‘BuzzerScreen.js’ with the help of student inputs.	

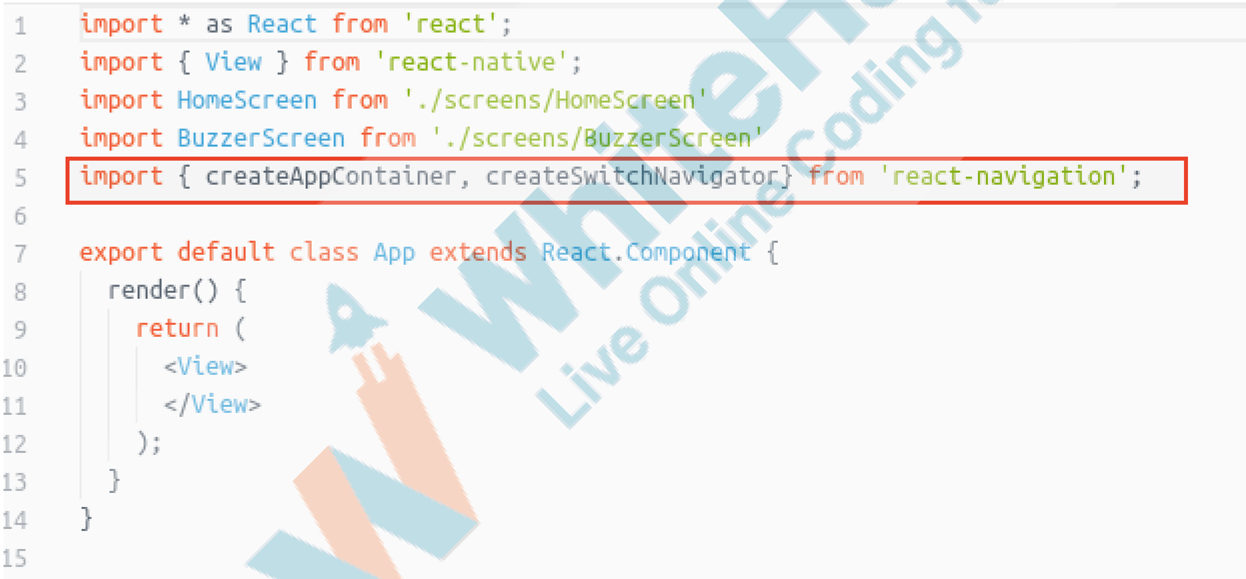
~~HomeScreen~~ BuzzerScreen in the image below



```
import * as React from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader';
import SoundButton from '../components/SoundButton';

export default class HomeScreen extends React.Component {
  render() {
    return (
      <View>
        <AppHeader />
        <SoundButton />
      </View>
    );
  }
}

```

	Alright. Now that we have the two screens, we will import them inside our App.js file. This is where we are going to use our screen. We will need to import two more components in our App.js file - ● ‘createSwitchNavigator’: It will allow us to create our AppNavigator which we will use to navigate the screens. ● ‘createAppContainer’: It holds the two screens and our App Navigator together.	The student listens, observes and asks questions.

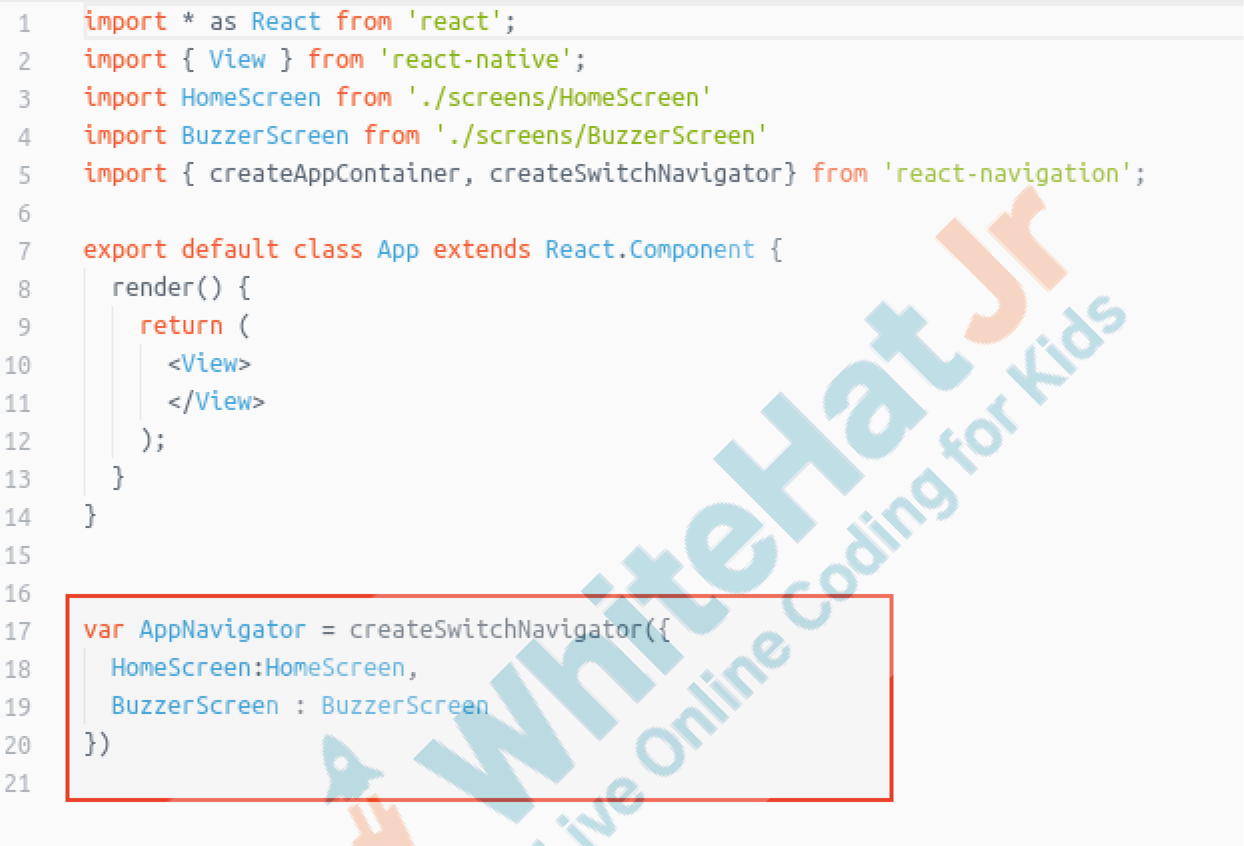
	Teacher imports ‘createSwitchNavigator’ and ‘createAppContainer’ from the 'react-navigation' library. Note 1: For US students kindly changed the version of react-navigation to 4.0.1 manually in the package.json Note 2: These are inside curly brackets. These are also case-sensitive	



```
import * as React from 'react';
import { View } from 'react-native';
import HomeScreen from './screens/HomeScreen'
import BuzzerScreen from './screens/BuzzerScreen'
import { createAppContainer, createSwitchNavigator} from 'react-navigation';

export default class App extends React.Component {
  render() {
    return (
      <View>
      </View>
    );
  }
}
```

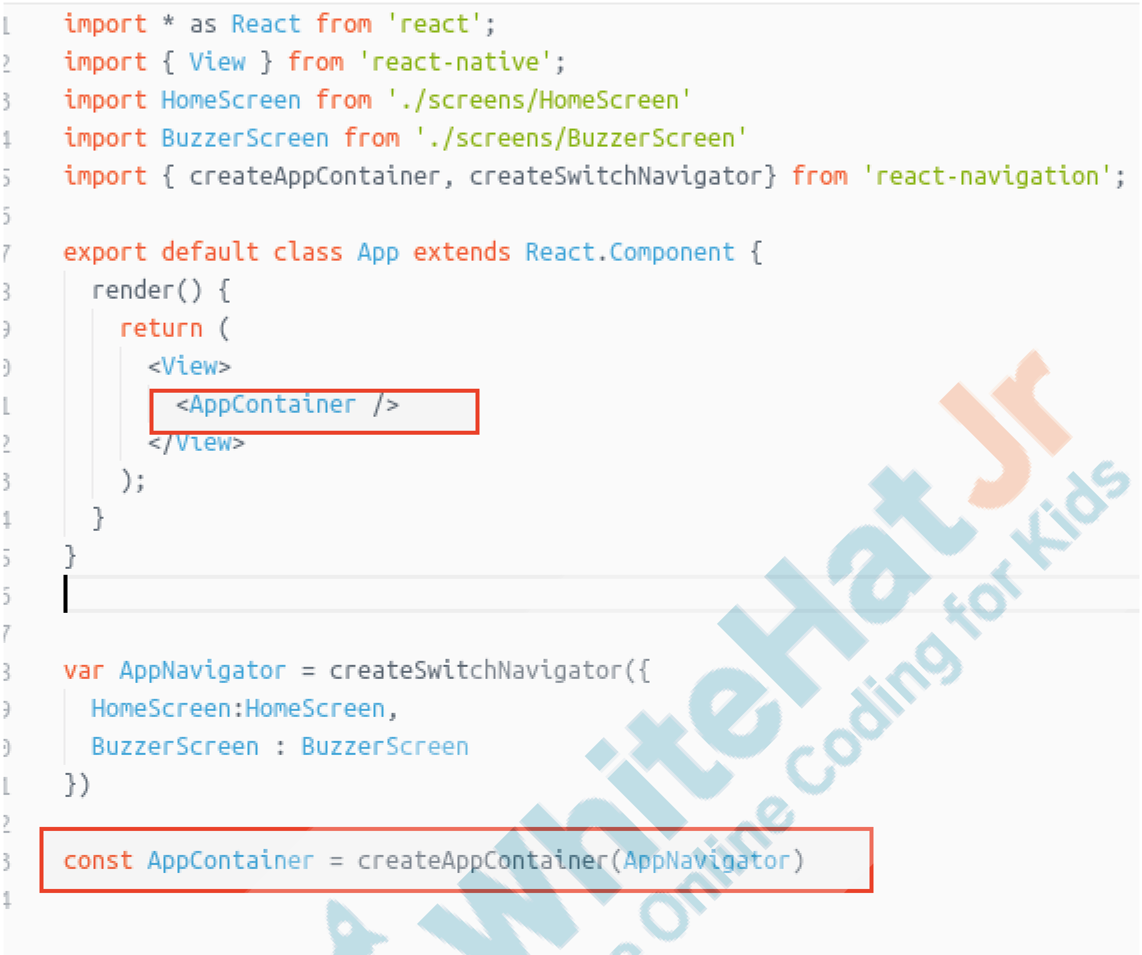
	Now. let's use ‘createSwitchNavigator’ to create our AppNavigator. ‘createSwitchNavigator’ takes a JSON object as an argument. A JSON object contains key names and the values corresponding to key names. This JSON contains the list of screens and their key names.	The student listens, observes and asks questions.

	Teacher writes code to create the AppNavigator	



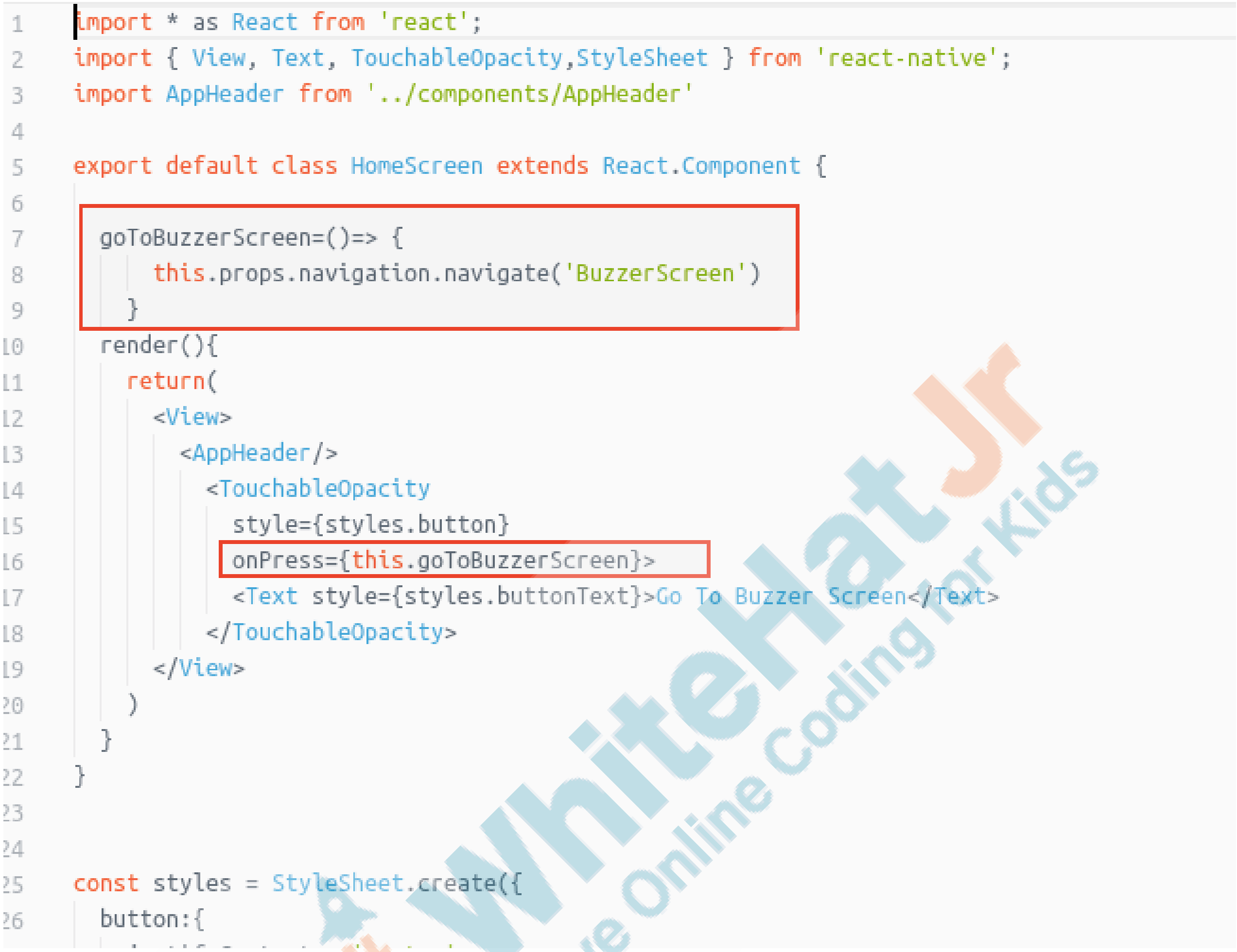
```
import * as React from 'react';
import { View } from 'react-native';
import HomeScreen from './screens/HomeScreen'
import BuzzerScreen from './screens/BuzzerScreen'
import { createAppContainer, createSwitchNavigator} from 'react-navigation';

export default class App extends React.Component {
  render() {
    return (
      <View>
      </View>
    );
  }
}


var AppNavigator = createSwitchNavigator({
  HomeScreen:HomeScreen,
  BuzzerScreen : BuzzerScreen
})

```

	Now, let's quickly create our AppContainer. Teacher writes code to create the AppContainer using ‘createAppContainer’. We can render the AppContainer inside our App class.	The student listens, observes and asks questions



```
import * as React from 'react';
import { View } from 'react-native';
import HomeScreen from './screens/HomeScreen'
import BuzzerScreen from './screens/BuzzerScreen'
import { createAppContainer, createSwitchNavigator} from 'react-navigation';

export default class App extends React.Component {
  render() {
    return (
      <View>
        <AppContainer />
      </View>
    );
  }
}


var AppNavigator = createSwitchNavigator({
  HomeScreen:HomeScreen,
  BuzzerScreen : BuzzerScreen
})

const AppContainer = createAppContainer(AppNavigator)
```

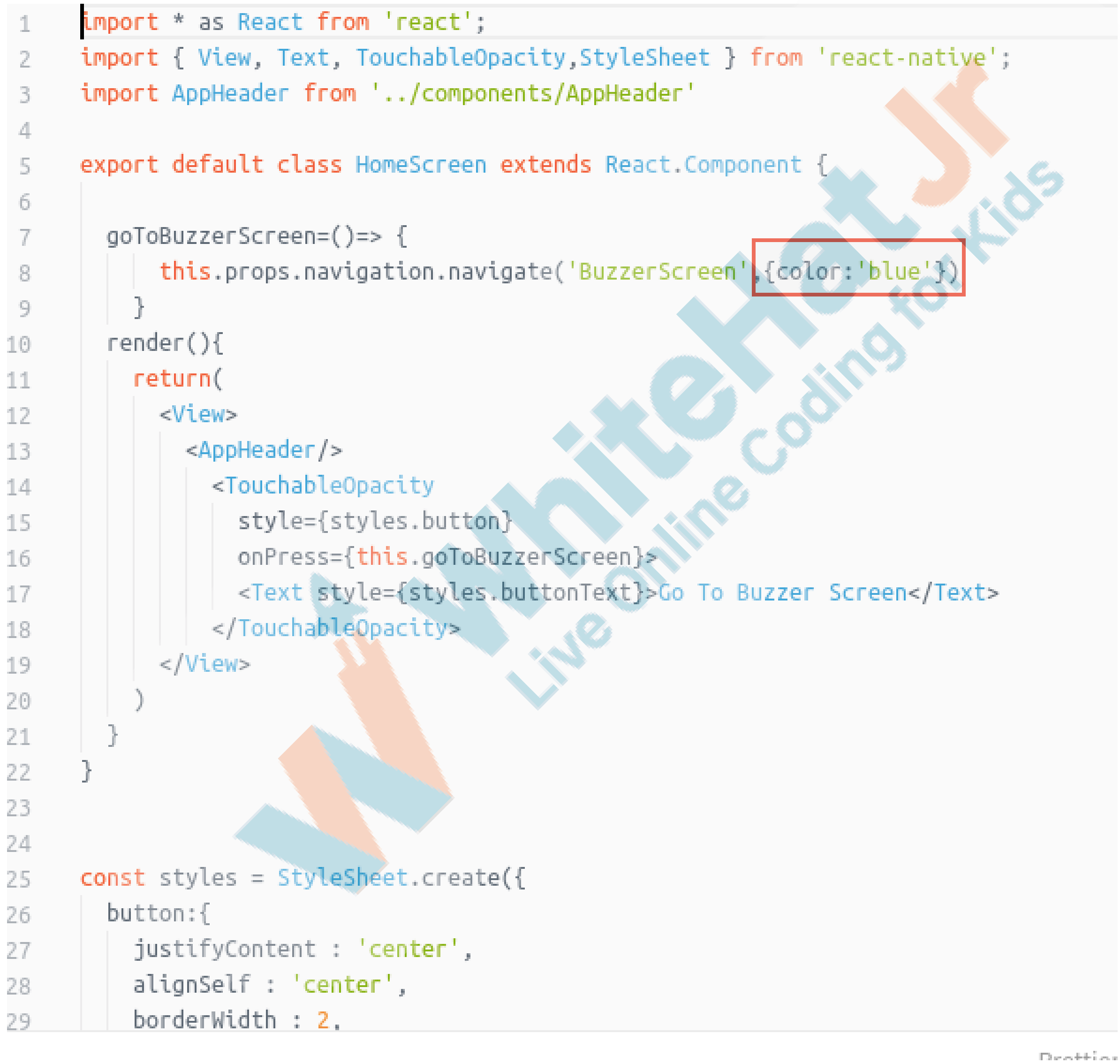
	Now, we have only one thing left before we can actually use our Switch Navigator! Can you recall what is that?	ESR: The function which will be called when we press the Button on the HomeScreen.
	Teacher shows how to navigate on pressing the button using 'props.navigation.navigate()' function inside HomeScreen. We need to pass the key name of the screen as an argument to 'props.navigation.navigate()'	The student observes and asks questions.



```
import * as React from 'react';
import { View, Text, TouchableOpacity,StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader'

export default class HomeScreen extends React.Component {

  goToBuzzerScreen=()=> {
    this.props.navigation.navigate('BuzzerScreen')
  }
  render(){
    return(
      <View>
        <AppHeader/>
          <TouchableOpacity
            style={styles.button}
            onPress={this.goToBuzzerScreen}>
            <Text style={styles.buttonText}>Go To Buzzer Screen</Text>
          </TouchableOpacity>
      </View>
    )
  }
}


const styles = StyleSheet.create({
  button:{
```

	Awesome! We are almost done. Let's quickly test if our app works!	Teacher and the student test the Expo App on Android/iOS by scanning the QR code.
	Amazing! One more interesting thing. You can also pass the data from one screen to another by passing it as an argument to 'props.navigation.navigate()'. The data is passed as an object (JSON) with key names.	The student observes and learns.

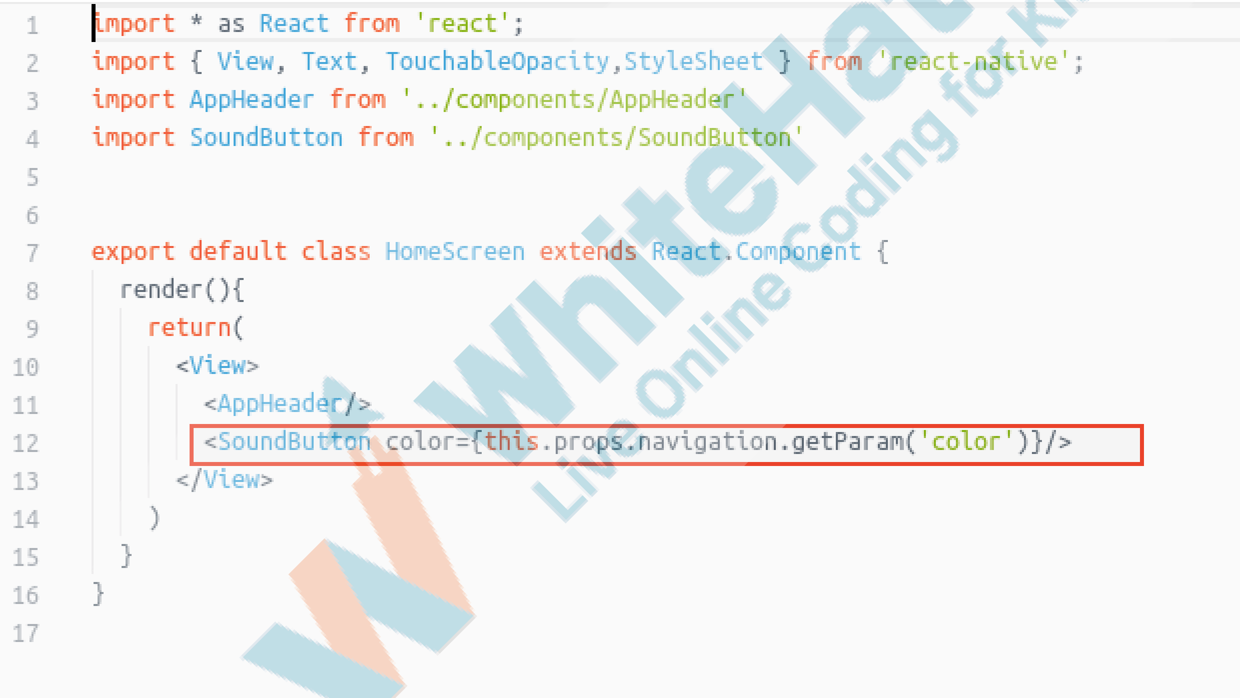
	Let's pass some random color to BuzzerScreen which we will use to create the color of the SoundButton. Teacher shows how to pass data from HomeScreen to BuzzerScreen.	



```
import * as React from 'react';
import { View, Text, TouchableOpacity,StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader'

export default class HomeScreen extends React.Component {

  goToBuzzerScreen=()=> {
      this.props.navigation.navigate('BuzzerScreen',{color:'blue'})
    }
  render(){
    return(
      <View>
        <AppHeader/>
          <TouchableOpacity
            style={styles.button}
            onPress={this.goToBuzzerScreen}>
            <Text style={styles.buttonText}>Go To Buzzer Screen</Text>
          </TouchableOpacity>
      </View>
    )
  }
}


const styles = StyleSheet.create({
  button:{
    justifyContent : 'center',
    alignSelf : 'center',
    borderWidth : 2,
```

	We can now use the data in the BuzzerScreen using the key name from the data.	The student observes and learns.

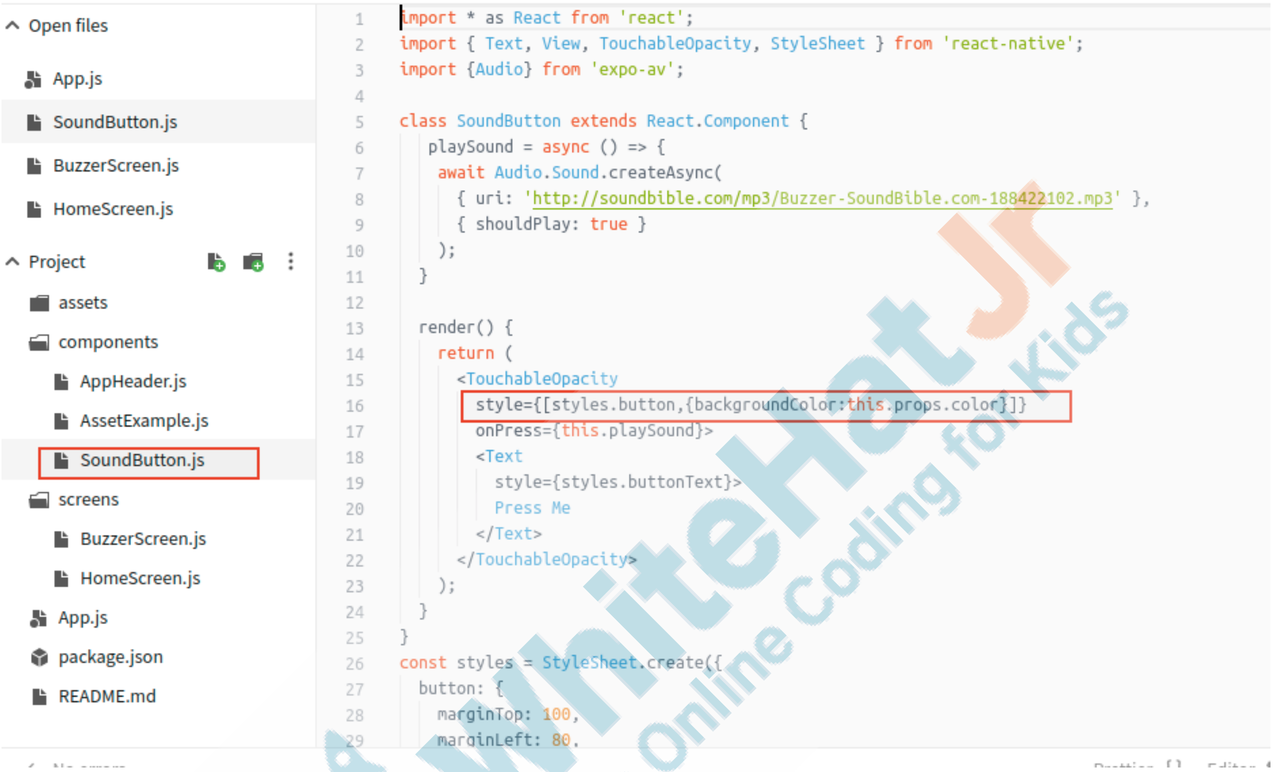
	Let’s create a prop called ‘color’ in our ‘SoundButton’. We can pass the data from HomeScreen to this prop. Teacher shows how to use the color data from HomeScreen in BuzzerScreen using this.props.navigation.getParam(‘color’)	



```
import * as React from 'react';
import { View, Text, TouchableOpacity,StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader'
import SoundButton from '../components/SoundButton'


export default class HomeScreen extends React.Component {
  render(){
    return(
      <View>
        <AppHeader/>
        <SoundButton color={this.props.navigation.getParam('color')}/>
      </View>
    )
  }
}

```

	Teacher shows how to use the color prop in the ‘SoundButton’ component. **Note**: Additional styling properties should be passed in the style prop inside an array.	The student observes and learns.

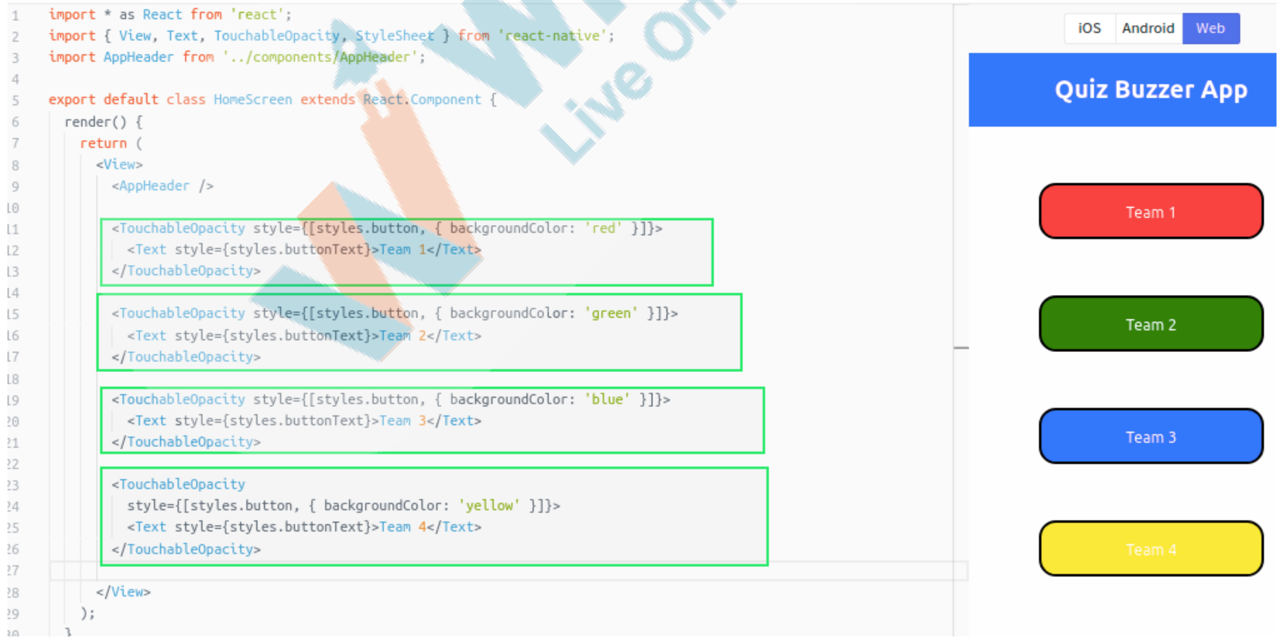
	‘styles.button’ and ‘{backgroundColor: this.props.color}’ are objects passed to style prop inside an array.	



```
import * as React from 'react';
import { Text, View, TouchableOpacity, StyleSheet } from 'react-native';
import {Audio} from 'expo-av';

class SoundButton extends React.Component {
  playSound = async () => {
   await Audio.Sound.createAsync(
    { uri: 'http://soundbible.com/mp3/Buzzer-SoundBible.com-188422102.mp3' },
    { shouldPlay: true }
   );
  }

  render() {
    return (
      <TouchableOpacity
        style={[styles.button,{backgroundColor:this.props.color}]}
        onPress={this.playSound}>
        <Text
          style={styles.buttonText}>
          Press Me
        </Text>
      </TouchableOpacity>
    );
  }
}
const styles = StyleSheet.create({
  button: {
    marginTop: 100,
    marginLeft: 80,
```

	Let's test our application and run it to see if it works.	The student also runs the code and tests it.
	Great work! Now time for a challenge for you. Why don't you create a two screen app where the user chooses their team by pressing a button on the home screen. The user then sees the SoundButton in the buzzer app with the color of their team. Do you think you can do that?	ESR: Yes!

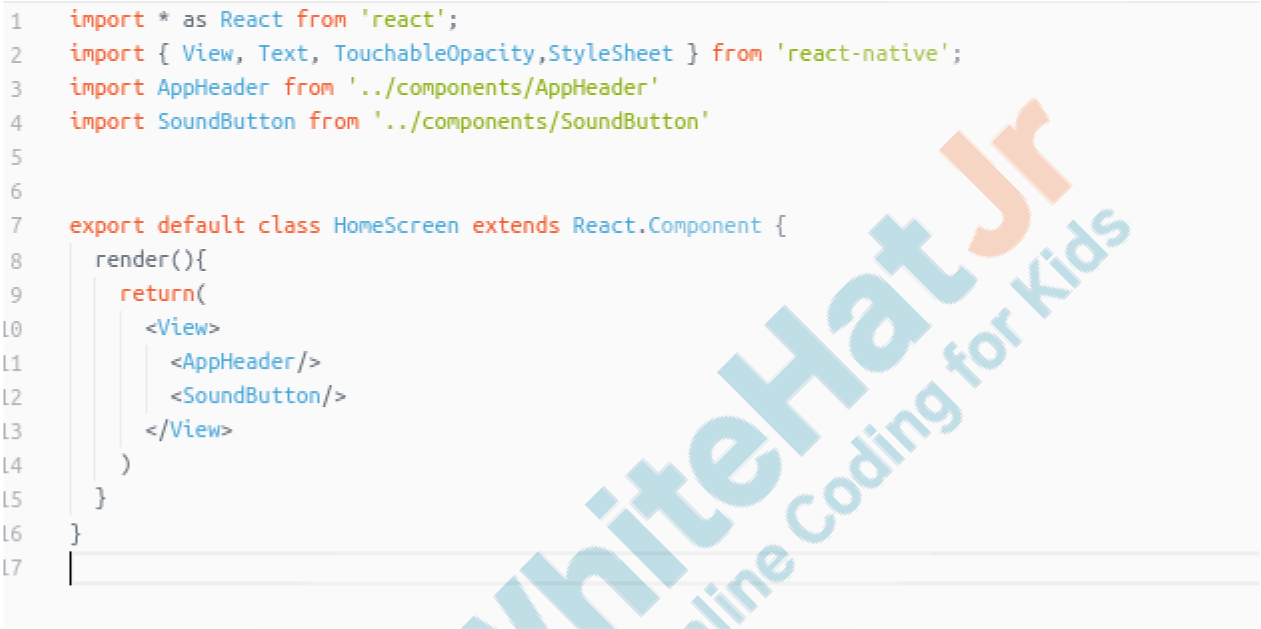
	Great. Let's get started then.	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
• Ask Student to press ESC key to come back to panel **• Guide Student to start Screen Share** **• Teacher gets into Fullscreen**		
<u>ACTIVITY</u> **• Create a screen which allows users to select their team through buttons of different colors.** **• Navigate to the Buzzer screen where the Buzzer is of the color of the team.**		
Step 3: Student-Led Activity (15 min)	Guide the student to create the HomeScreen.	The student creates a HomeScreen which contains 4 different colored buttons - red, green, blue and yellow.



```
import * as React from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader';

export default class HomeScreen extends React.Component {
  render() {
    return (
      <View>
        <AppHeader />

        <TouchableOpacity style={[styles.button, { backgroundColor: 'red' }]}>
          <Text style={styles.buttonText}>Team 1</Text>
        </TouchableOpacity>

        <TouchableOpacity style={[styles.button, { backgroundColor: 'green' }]}>
          <Text style={styles.buttonText}>Team 2</Text>
        </TouchableOpacity>

        <TouchableOpacity style={[styles.button, { backgroundColor: 'blue' }]}>
          <Text style={styles.buttonText}>Team 3</Text>
        </TouchableOpacity>

        <TouchableOpacity
          style={[styles.button, { backgroundColor: 'yellow' }]}>
          <Text style={styles.buttonText}>Team 4</Text>
        </TouchableOpacity>

      </View>
    );
  }
```

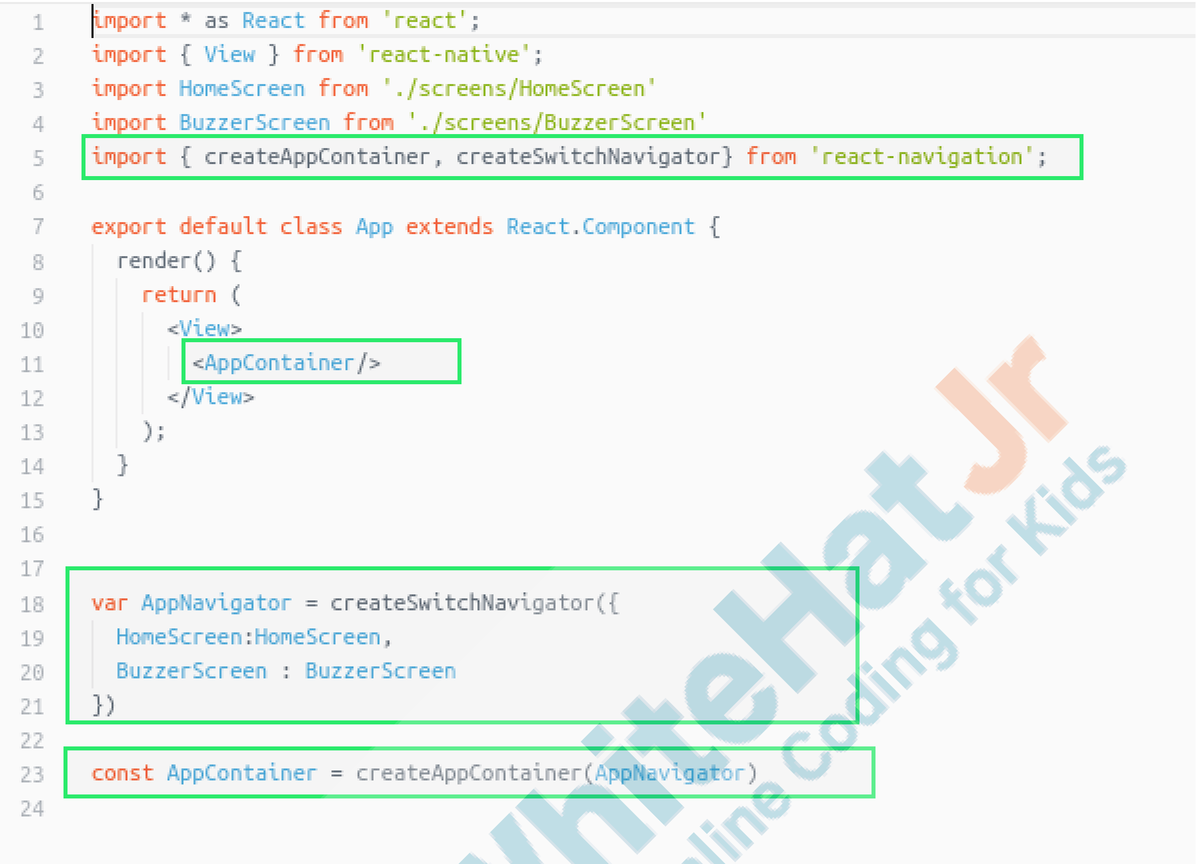
<table>
<tr><td></td><td>Guid the student to create the Buzzer Screen.</td><td>The student creates a BuzzerScreen / HomeScreen which holds the SoundButton.</td></tr>
<tr><td colspan="3">

~~HomeScreen~~ BuzzerScreen in the image below



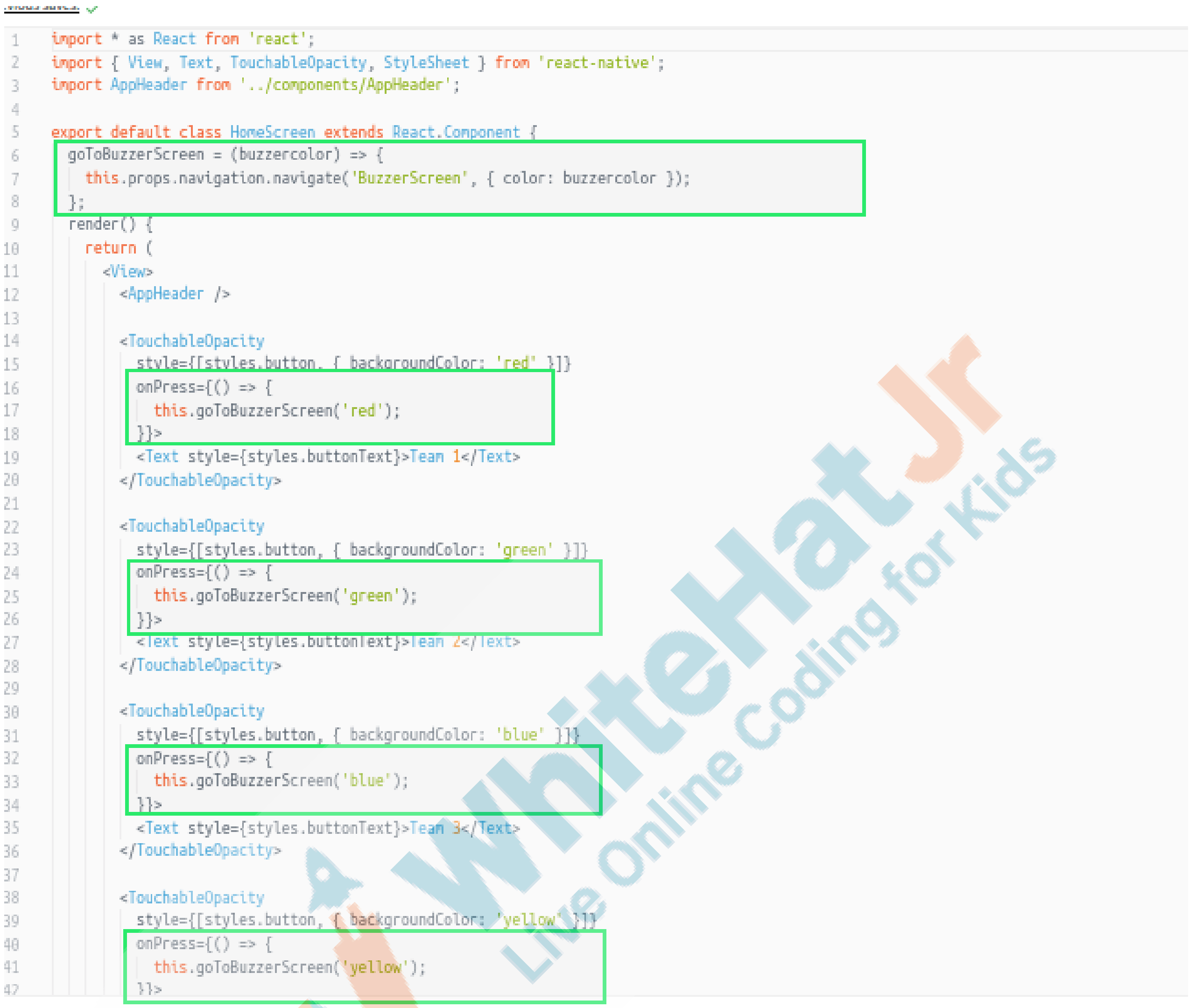
```
import * as React from 'react';
import { View, Text, TouchableOpacity,StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader'
import SoundButton from '../components/SoundButton'


export default class HomeScreen extends React.Component {
  render(){
    return(
      <View>
        <AppHeader/>
        <SoundButton/>
      </View>
    )
  }
}

```

</td></tr>
<tr><td></td><td>Guide the student to create AppNavigator containing both the screens.
Guide the student to create an AppContainer containing the AppNavigator.</td><td>The student creates an AppNavigator and AppContainer in App.js file.</td></tr>
</table>



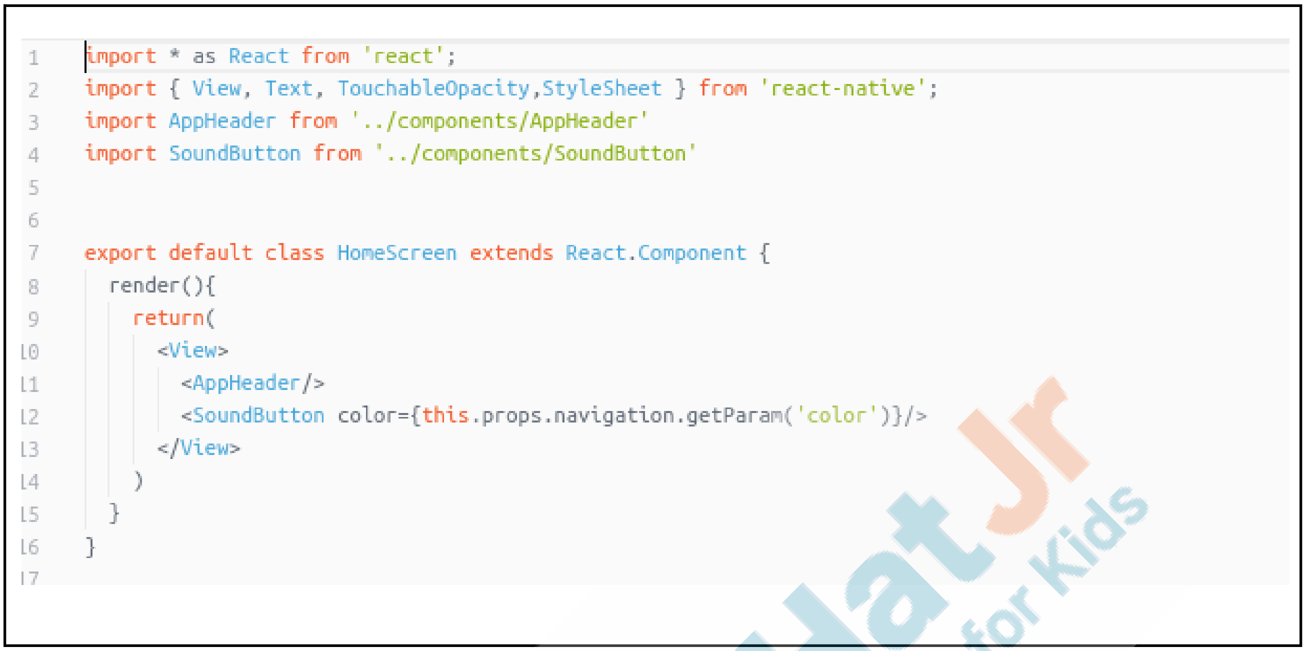
```
import * as React from 'react';
import { View } from 'react-native';
import HomeScreen from './screens/HomeScreen'
import BuzzerScreen from './screens/BuzzerScreen'
import { createAppContainer, createSwitchNavigator} from 'react-navigation';

export default class App extends React.Component {
  render() {
    return (
      <View>
        <AppContainer/>
      </View>
    );
  }
}


var AppNavigator = createSwitchNavigator({
  HomeScreen:HomeScreen,
  BuzzerScreen : BuzzerScreen
})

const AppContainer = createAppContainer(AppNavigator)

```

	Guide the student to create functions to navigate to a different screen and pass color data through onPress prop of the Button.	The Student writes a function to be called under onPress for each of the Buttons in their app. The function navigates the user to the BuzzerScreen and passes the color data.

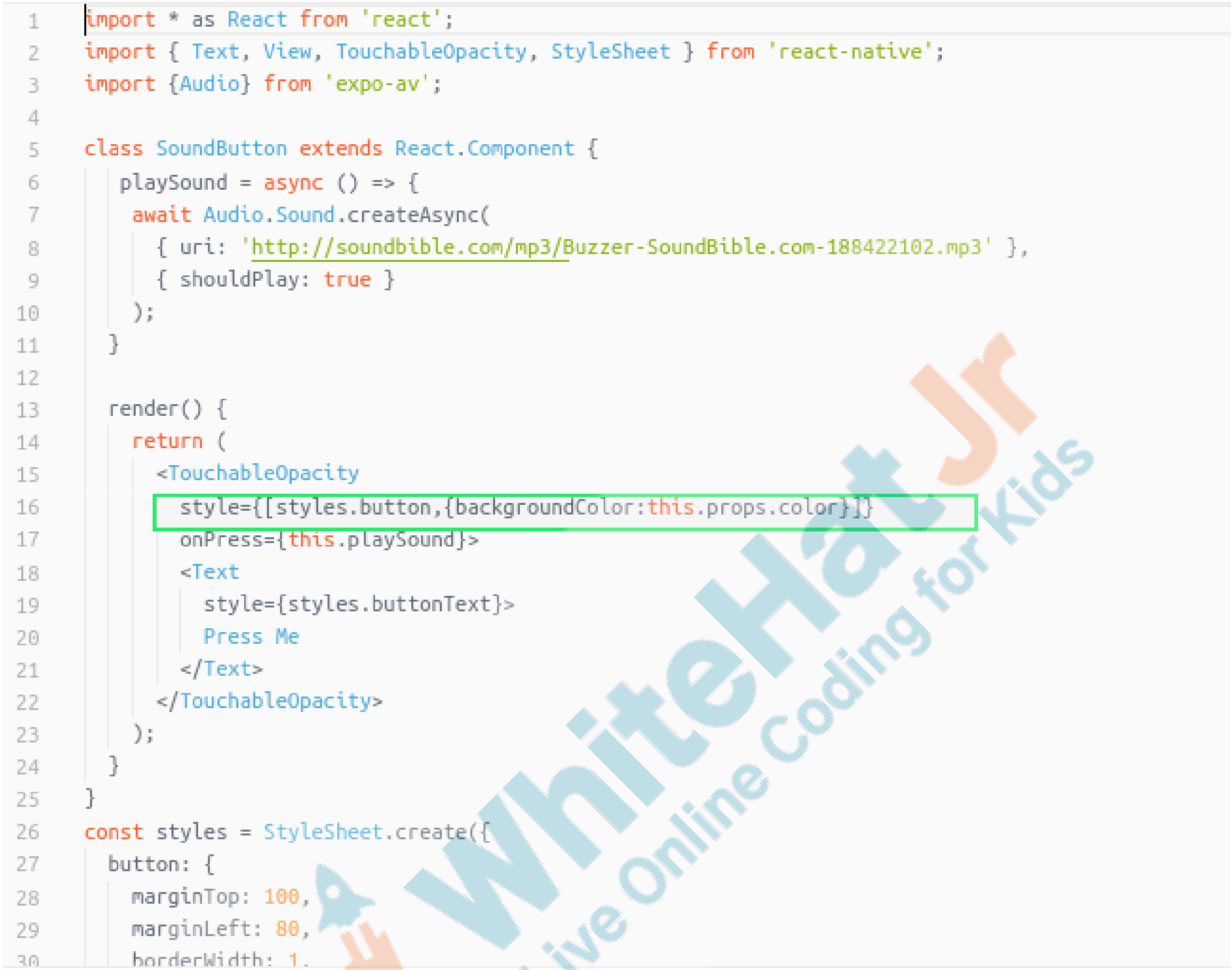


```
import * as React from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader';

export default class HomeScreen extends React.Component {
  goToBuzzerScreen = (buzzercolor) => {
    this.props.navigation.navigate('BuzzerScreen', { color: buzzercolor });
  };
  render() {
    return (
      <View>
        <AppHeader />

        <TouchableOpacity
          style={[styles.button, { backgroundColor: 'red' }]}
          onPress={() => {
            this.goToBuzzerScreen('red');
          }}>
          <Text style={styles.buttonText}>Team 1</Text>
        </TouchableOpacity>

        <TouchableOpacity
          style={[styles.button, { backgroundColor: 'green' }]}
          onPress={() => {
            this.goToBuzzerScreen('green');
          }}>
          <Text style={styles.buttonText}>Team 2</Text>
        </TouchableOpacity>

        <TouchableOpacity
          style={[styles.button, { backgroundColor: 'blue' }]}
          onPress={() => {
            this.goToBuzzerScreen('blue');
          }}>
          <Text style={styles.buttonText}>Team 3</Text>
        </TouchableOpacity>

        <TouchableOpacity
          style={[styles.button, { backgroundColor: 'yellow' }]}
          onPress={() => {
            this.goToBuzzerScreen('yellow');
          }}>
```

	Guide the student to use the color data in BuzzerScreen.	The student uses the color data in BuzzerScreen to style the color of the buzzer Button.



```
import * as React from 'react';
import { View, Text, TouchableOpacity,StyleSheet } from 'react-native';
import AppHeader from '../components/AppHeader'
import SoundButton from '../components/SoundButton'


export default class HomeScreen extends React.Component {
  render(){
    return(
      <View>
        <AppHeader/>
        <SoundButton color={this.props.navigation.getParam('color')}/>
      </View>
    )
  }
}
```



```
import * as React from 'react';
import { Text, View, TouchableOpacity, StyleSheet } from 'react-native';
import {Audio} from 'expo-av';

class SoundButton extends React.Component {
   playSound = async () => {
    await Audio.Sound.createAsync(
      { uri: 'http://soundbible.com/mp3/Buzzer-SoundBible.com-188422102.mp3' },
      { shouldPlay: true }
    );
  }

  render() {
    return (
      <TouchableOpacity
        style={[styles.button,{backgroundColor:this.props.color}]}
        onPress={this.playSound}>
        <Text
          style={styles.buttonText}>
          Press Me
        </Text>
      </TouchableOpacity>
    );
  }
}
const styles = StyleSheet.create({
  button: {
    marginTop: 100,
    marginLeft: 80,
    borderWidth: 1,
```

	Help the student run the app and test it.	The student runs and tests the app on their phone.

Teacher Guides Student to Stop Screen Share

FEEDBACK

- **Encourage the student to create more screens and navigate between them using switch navigator.**
- **Encourage the student to make reflection notes in the markdown format.**
- **Complement the student for her/his effort in the class.**

Step 4: **Wrap-Up** **(5 min)**	Great!	

	We are inching closer to creating our Buzzer App. The user can choose their teams in your app and they can play the Buzzer Sound. What else do we want in our app now?	We want to detect who pressed the button first.
	Yes and for that we will be using databases. Remember what database we used in our Car Racing game ?	ESR: Firebase Realtime Database
	Correct! We will be doing that in the next class. Before we close this class, can we quickly review what we learned in today's class?	ESR: - We learned how to create Switch Navigation between the two screens. - We also learned how to pass data from one screen to another.
	Awesome! In the next class, we will identify who pressed the button first and our app will be almost done. You get a “hats off”. Looking forward to the next class.	Make sure you have given at least 2 Hats Off during the class for: Creatively Solved Activities +10 Great Question +10 Strong Concentration +10

Project Pointers and Cues (5 min)	***This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.** Knowledge App **Goal of the Project:** In class 57, you learned about “Switch Navigation”. You coded for a multi-screen app where you could press a button to navigate to a different screen. You also learned how to pass data from one screen to another screen using props. In this project, you will have to apply what you have learned in the class and create an error free Knowledge App. **Story:** Gazala loves reading interesting facts and she always shares it with others. So she started creating an application where she can share facts about different fields. You have to help Gazala in finding and fixing the errors in her Knowledge App. I am very excited to see your project solution and I know you both will do really well. Bye Bye!	

Teacher Clicks	✖ End Class	
Additional Activities	Encourage the student to create more screens and navigate between them using switch navigator.	The student creates more screens and creates switch navigation between them.
	Encourage the student to write reflection notes in their reflection journal using markdown. Use these as guiding questions: ● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	The student uses the markdown editor to write her/his reflection in a reflection journal.

Activity	Activity Name	Links
Teacher Activity 1	Switch Navigation	https://snack.expo.io/@rajeevtfi/student-activity-1-app-header-reference
Teacher Activity 2	Teacher Reference -1	https://snack.expo.io/@rajeevtfi/teacher-activity-1-reference:-switch-navigator
Student Activity 1	Switch Navigation	https://snack.expo.io/@rajeevtfi/student-activity-1-app-header-reference
Teacher Activity 3	Teacher Reference -2	https://snack.expo.io/@rajeevtfi/student-activity-1:-switch-navigator-reference

Project Solution	Knowledge App	https://snack.expo.io/@snerrus/5f1f7de7b33be43e5cc98dfc4940f6b1