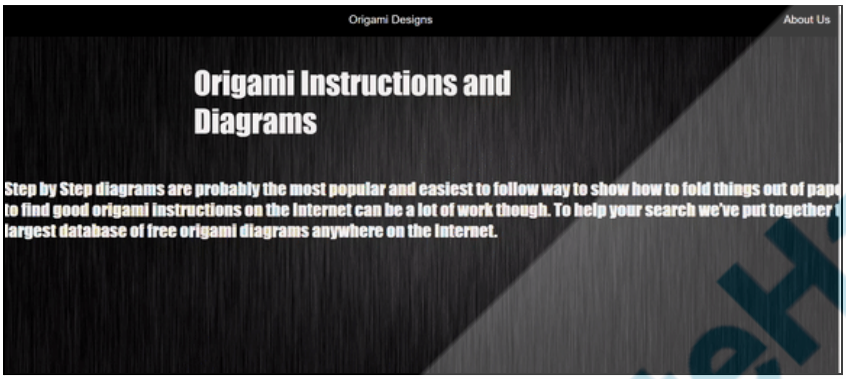
Topic	React Philosophy	
Class Description	**Students will compare the two languages - HTML and Javascript. Students will be introduced to the React Philosophy and the new JSX language, which they will be using to code mobile apps in React Native. Students will create a simple React Native component for their application.**	
Class	**C53**	
Class time	**45 mins**	
Goal	• Compare HTML (declarative language) and Javascript (Imperative language). • Introduction to React philosophy. • Creating a simple React Native component for their React Native app in expo.	
Resources Required	• Teacher Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Android/iOS Smartphone with Expo App installed ○ Expo Account Login • Student Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Android/iOS Smartphone with Expo App installed ○ Expo Account Login	
Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 min** **5 min**

WARM-UP SESSION - 5 mins

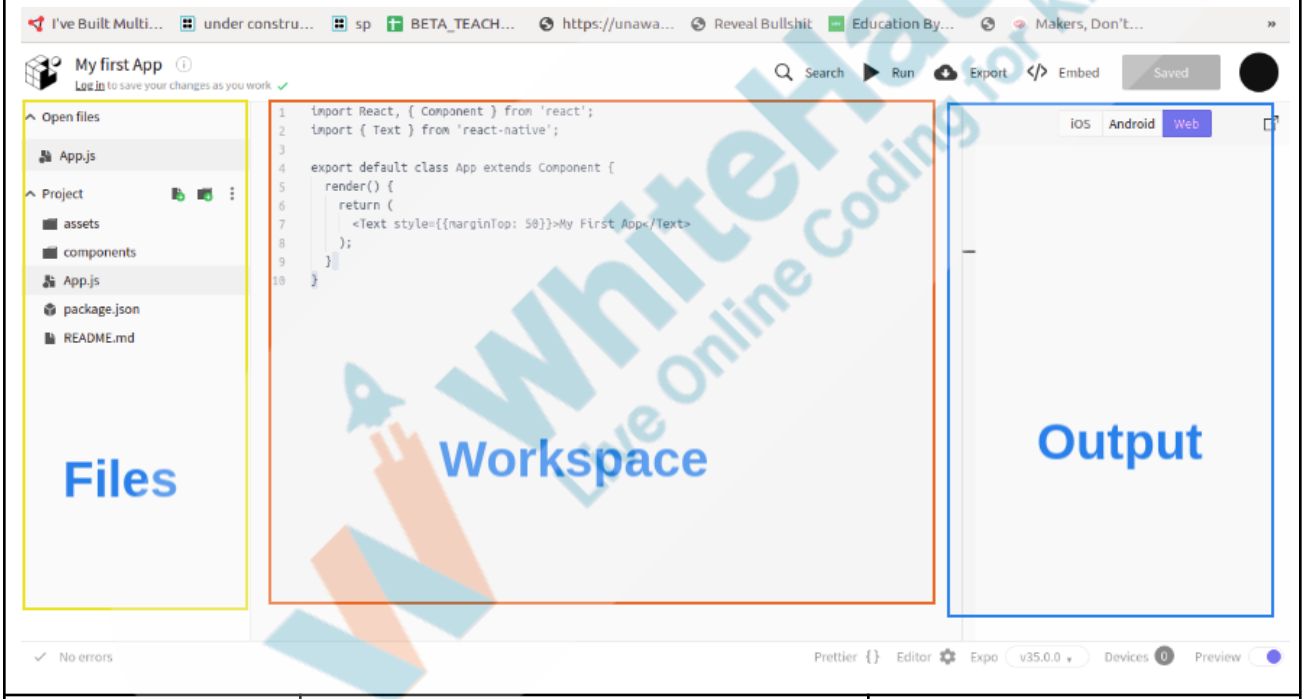
CONTEXT

- **Differentiate between HTML and Javascript.**
- **Introduce JSX for coding apps using React Native.**

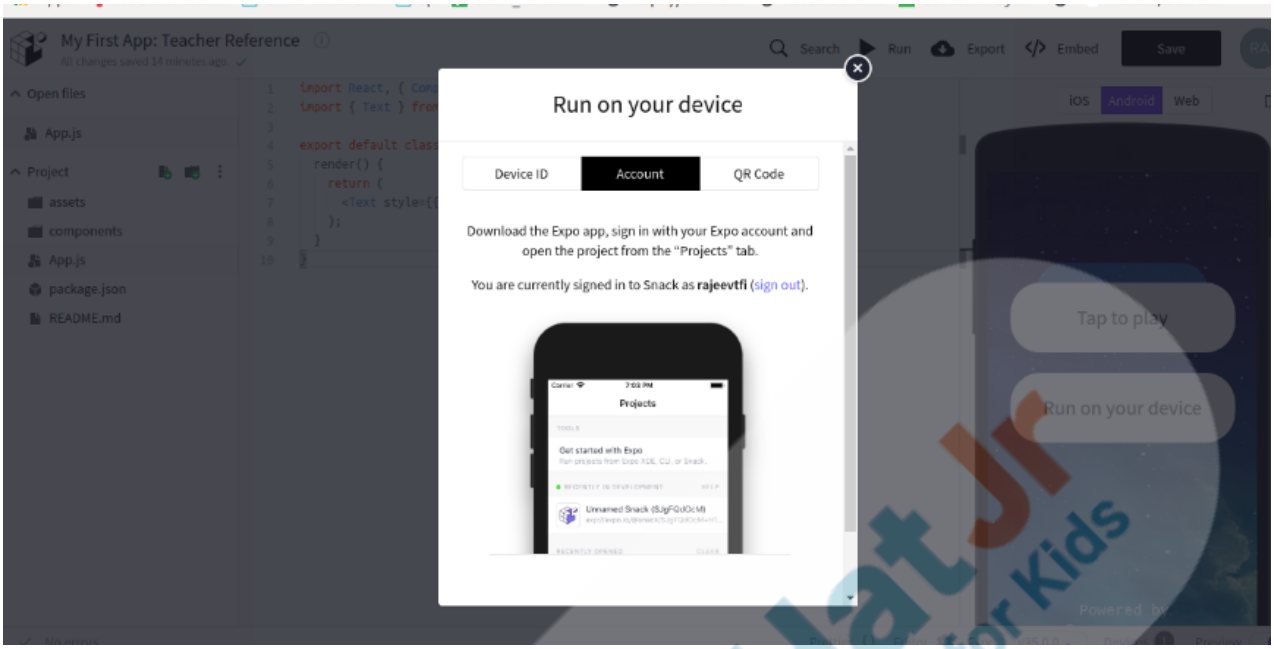
Teacher starts slideshow from slides 1 to 14 Refer to speaker notes and follow the instructions on each slide.	
Activity details	**Solution/Guidelines**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Run the presentation from slide 1 to slide 3** **Following are the WARM-UP session deliverables:** • **Greet the student.** • **Revision of previous class activities.** • **Quizzes**	**ESR**: Hi, thanks, Yes I am excited about it! Click on the slide show tab and present the slides
QnA Session	
Question	**Answer**
Identify the option that can update the color of the links that have been visited (clicked) to white. Hexadecimal code of white is #FFFFFF. A. B. C.	**D.**

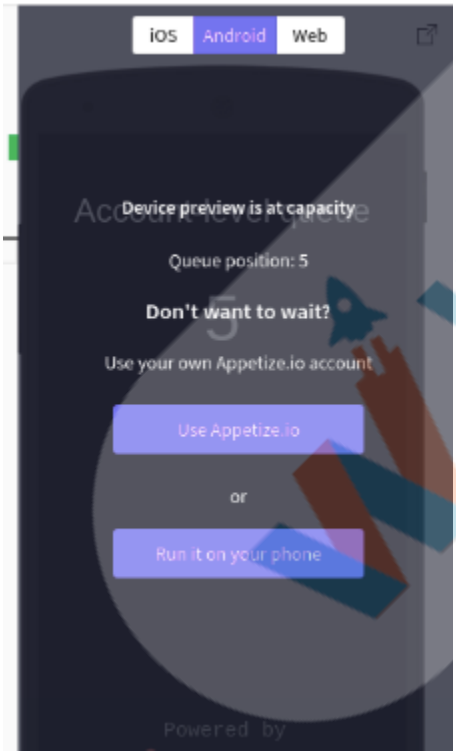
D. ```a:visited { color: #ffffff; }```	
Identify the option to amplify the size of an image when the mouse hovers over that image as shown below. Image belongs to the origami class.  A. ```.origami:hover{ transform: scale(1.2); }``` B. ```.origami : mouseover{ transform: scale(1.2); }``` C. ```.origami:link{ transform: scale(1.2); }``` D. ```origami:hover{ transform: scale(1.2); }```	**A**
Continue the WARM-UP session	
Activity details	**Solution/Guidelines**

Run the presentation from slide 4 to slide 14 to set the problem statement. **Following are the WARM-UP session deliverables:** • Appreciate the student. • Explain React Native		Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.
Teacher ends slideshow 		
TEACHER-LED ACTIVITY - 15 mins		
Teacher Initiates Screen Share		
CHALLENGE • **Create a simple React Native Component.**		
Step 2: Teacher-led Activity (15 min)	We will be writing JSX and React Native code in an online editor called 'snack'. 'Snack' is an online editor written by Expo, which allows us to write and compile React Native code in an online environment. We can also preview the output of the code using Snack. Note: Teacher is already logged in at Snack Expo. Teacher clicks on **Teacher Activity 1**	Student observes.
	The code you see might look intimidating but when we look into it, we will realize how simple and how powerful it is. Before we look into the code however, let us familiarize ourselves	The student looks at the environment to familiarize him/herself..

	with the coding environment. On the left side, you see all the files in the project. We will look into these files later. At the centre, you see the workspace where we will write our code. On the right side, you can see the output of your code. You can see the output for either Android, iOS or Web.	
		
	For Android/iOS, you can see the output either on your phone or on the android emulator in the browser. (By clicking Run on your device.) For Android/iOS, you can install Expo Client, sign in with the same account and open the project from under	The student checks the output of the code.

	Projects Tab. For Android, you can also scan the QR Code. (Encourage the student to scan the QR code from their Expo App and check the output.) For seeing the output on the device emulator in the browser, you have to press "Tap to play". You might have to wait in a queue as there are many systems online using a common emulator. Note: Teacher practically shows the different ways of seeing the output.	





	So, before we get into the code; can you tell me, what is the output you see?	ESR: I can see the text 'My First App' written on the screen.

	Let's look at the code now. What are the different parts of the code that you can see and understand? Do not worry we will get deeper into each part of the code.	ESR: The student makes an attempt to understand some parts of the code. For example: He/She can recognize the 'class' and 'extends' declaration. The student can also recognize HTML tag like feature in <Text></Text> He/She can recognize the marginTop style property from CSS.

	Great! Now let's look at each part of the code which we see. We have talked earlier about how complex apps are made up of smaller and simpler components. In React Native, every app is made up of simpler components. <Text></Text> tag which you see is such a React component. This component allows you to write text on the screen. I can modify what is inside it to change the text written on the screen. Teacher modifies the text written on the screen.	The student observes the change in output and asks questions.

	Text is a component which is already defined in a library called React Native. In line 2, you can see how we are importing the component Text from React Native Library. **Note:** When compared to HTML tags which start with lowercase letters, React native components start with uppercase.	The student observes, listens and asks questions.

	Component is a Base class (Parent class) from which all components are inherited from. Remember, Baseclass in Angry Birds? and how each of the classes - Bird, Pig, Box - extended the Base Class and inherited all its properties and functions? Component is a baseclass and all React components are inherited from this baseclass Component. Component class is defined in React library. You can see how React and Component library are imported from react library in line 1.	The student observes, listens and asks questions.

	In line 4 you can see how a new Class called App extends the Component class. (Ignore export default for now. We will come back to it sometime later.) App itself is a React Native Component. All React Native components must live inside the App Component. Things will become clearer soon, when we practically start making an App. Before we go further, can you explain what you have understood so far?	ESR: A React Native App is made up of several components. All components are inherited from the baseclass call Component (defined in react library.) App is the main component inside which all other components should live. Text is a component defined in React Native library which allows us to show text on screen.

Awesome! Very Good! You have been quite good to grasp the concepts so far.

Now you can see that there is only one defined function inside the App Class - it is called render()

'render()' function simply displays whatever components are returned by it.

Here, you can see that the <Text> component is returned. So, the <Text> component gets rendered or displayed on the screen.

Notice that all the code in the file is Javascript except the code inside render. This is JSX. It contains tags which correspond to React Components.

	We can write Javascript inside JSX using { } 'curly brackets' You can also see that the Text component has a style 'property' defined on it. Just like html tags have some properties defined on them (For example: <img> tag has src, width, height etc.) "style" property takes a json object {}. When we are rendering components using JSX tags (<Text>), we can write/execute Javascript inside the {}. You see two {{ }} because one {} says that we are going to write javascript code. The other {} is for the json object. Also, we write css properties in React Native in camel case (marginTop instead of margin-top) Again, things will become much clearer when we practically start doing things. But before we do that can you quickly recap what you can understand from the code?	ESR: render() function displays the returned components on the screen. Components can have properties defined on them - like Text has a property called style. We can write javascript inside JSX tags, inside { }.

	Awesome! Now you have some basic understanding of the displayed code. Let's try to do something, very simple. Let's try to place the text somewhere in the centre. That should not be tough, right? Can you help me do it?	ESR: No Yes. I can help.
	Teacher centres the text with the help of student input.	Student tells the teacher how to place the Text in the centre.

	Great! Now to become comfortable, why don't you try to use another component called Button. I will tell you the properties that the Button component has. So, are you ready?	ESR: Yes!
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

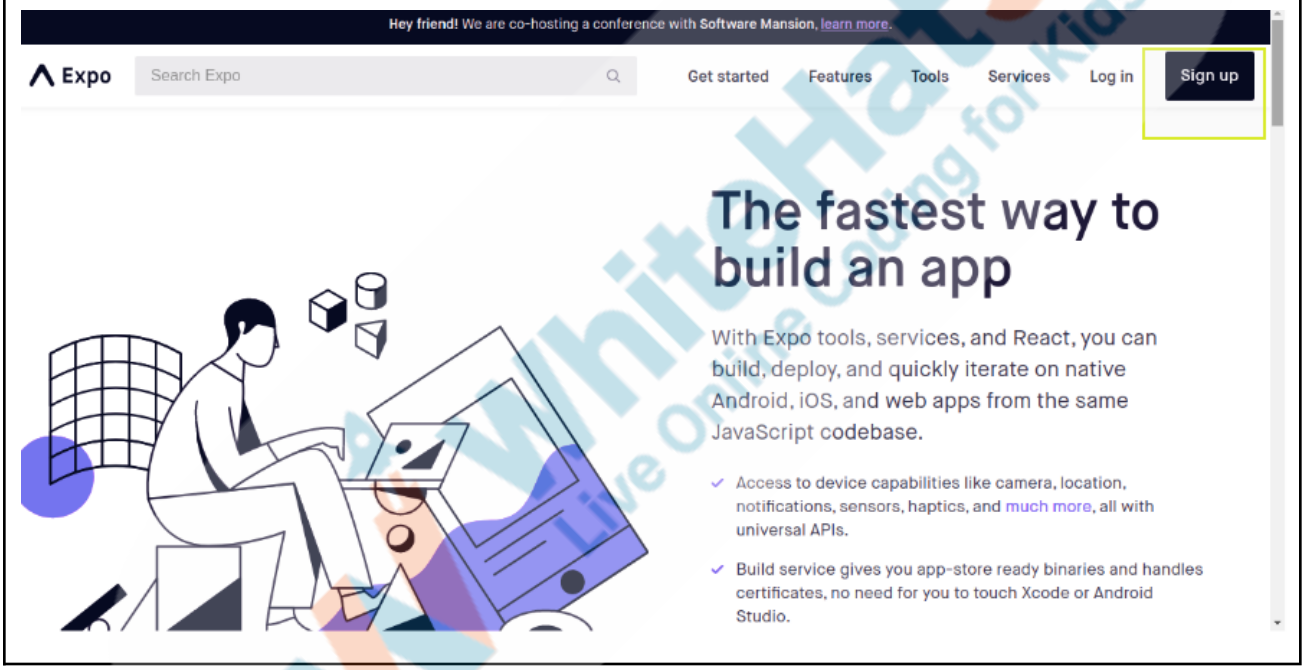
STUDENT-LED ACTIVITY - 25 mins

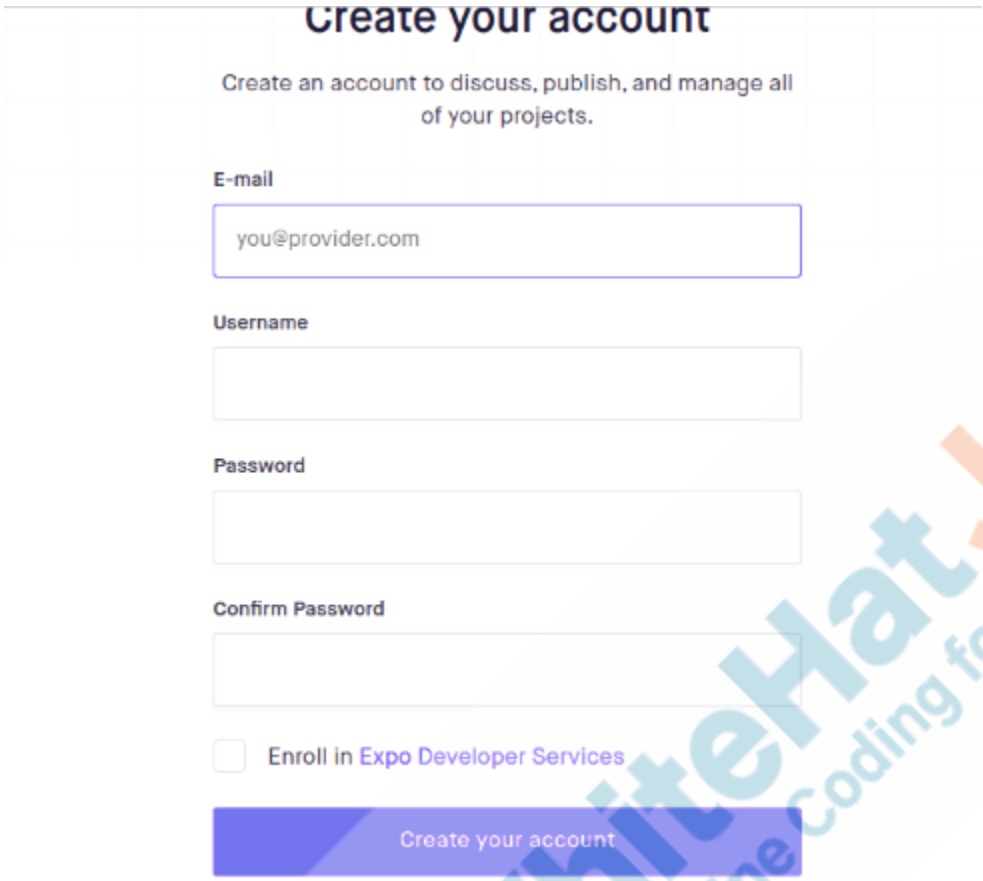
- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

ACTIVITY

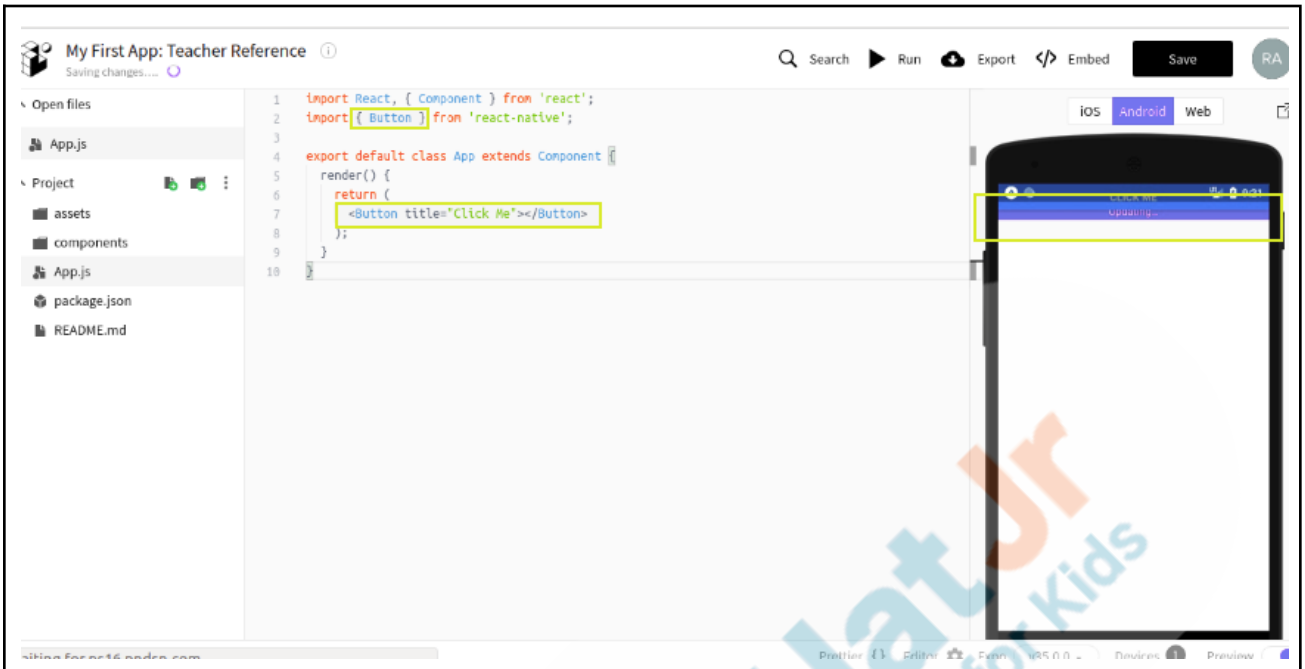
- **Define the class for the React native component.**
- **Use the React Native component in the app.**

<table>
<tr><td colspan="3">Teacher starts slideshow  :Slide 15 to 16
Refer to speaker notes and follow the instructions on each slide.</td></tr>
<tr><td>Step 3:
Student-Led Activity
(15 min)</td><td>Guide the student to create an Expo account and login.

Get the student to click on the <u>Student Activity Link 1</u></td><td>Student creates an Expo Account and clicks on <u>Student Activity Link 1</u></td></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

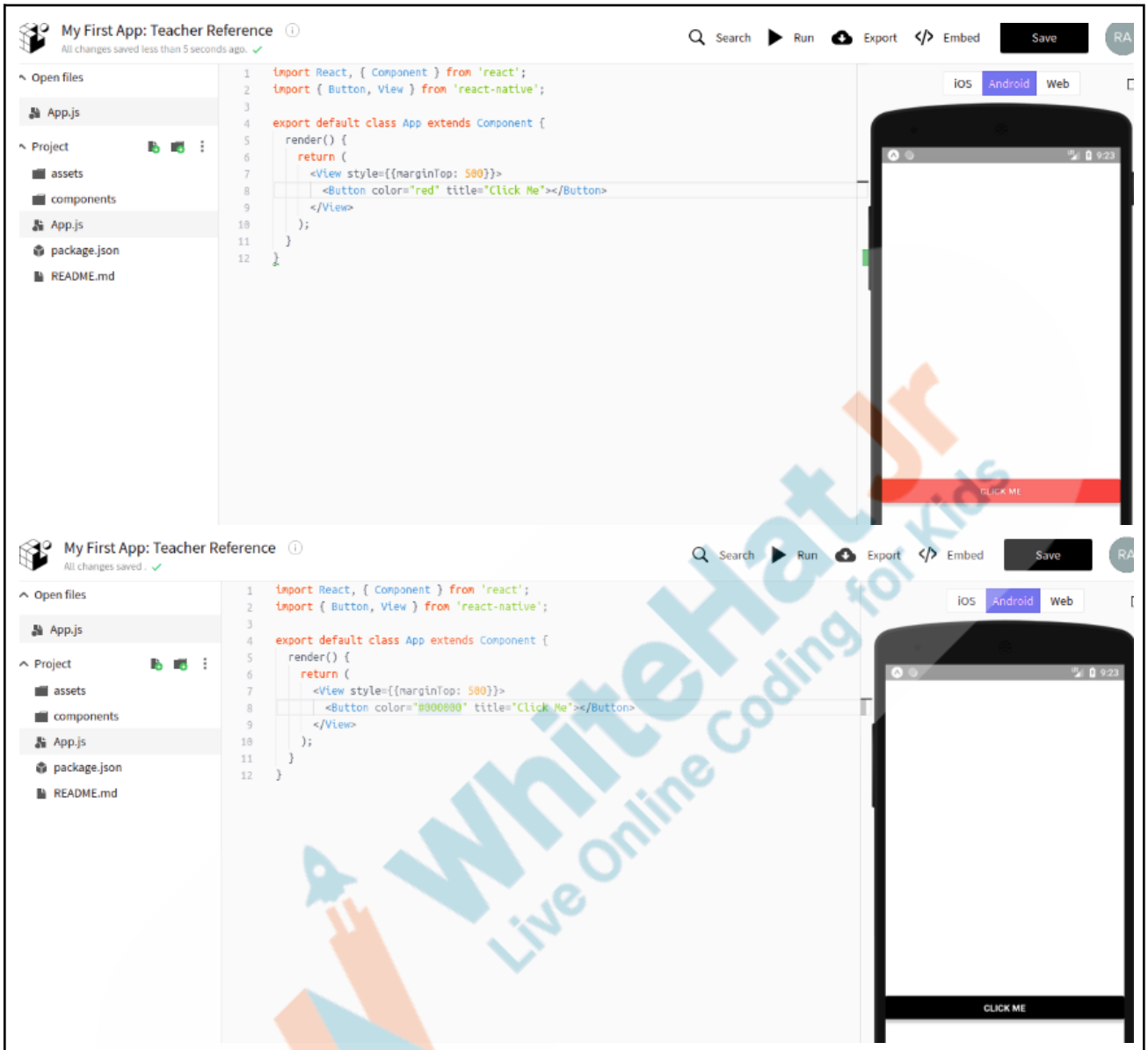


	Let's import Button instead of Text from react native library. We can use the <Button></Button> JSX tag to get a button component. <Button> has a property called 'title' which can be used to display text inside the button.	The student uses the <Button> component in place of <Text> component in the code. The student runs the code to see the output.



	You can see that there is a Button at the top of the app. It is half hidden by the top notification bar of the phone. How can we bring the Button down?	ESR: Using style property?
	Unfortunately, the button does not have a style property defined on it. Instead, we will use 'View Component' and place Button inside it. We can then adjust the style of 'View Component'. 'View Component' is simply an empty container - just like 'div' in html. It has a style property defined on it.	The student asks questions.

	Let's import ‘View Component’. Let's place the button inside the ‘View Component’ and add style to it so that the button is at the bottom.	The student imports the ‘View Component’ and places the ‘Button’ component inside it. He/She adds styling to the ‘View Component’ to bring the Button to the bottom.
		
	Very good. This is a great start! Button has another property called ‘color’ defined on it. You can use it to set a different color to the button. You can also use Hexadecimal numbers.	Student changes the color of the Button component.



	Great work so far. So now you know how to use components which are already defined in React Native. You have learned about three such components - Text, Button and View. There are many such react native components about which we will keep	ESR: yes!

	learning in the coming classes. Did we also say that you can create your own React Native component? Let's quickly learn how to do that.	
	Let's create a simple React Native component called 'RedButton'. This react native component will simply display a RedButton when rendered. We will need to create a class called RedButton which extends Component.	The student creates a class called 'RedButton' which extends Component.
		
	Now, we need to write a render function, which will return something. Let's quickly write a render function with an empty return statement.	The student creates a render function with an empty return function.

```
import React, { Component } from 'react';
import { Button, View } from 'react-native';

class RedButton extends Component{
  render(){
    return();
  }
}

export default class App extends Component {
  render() {
    return (
      <View style={{marginTop: 500}}>
        <Button color="#000000" title="Click Me"></Button>
      </View>
    );
  }
}
```

	We want to render a Red Button. We can write that in the return statement. Note: <Button/> is same as <Button></Button> The former is a self-enclosing tag since there is nothing inside <Button></Button> All tags in React native must be closed. Can you write the return statement for the Red Button? Guide the student where needed.	The student writes the return statement for rendering the RedButton. He/She adds the properties of color and title to the Button.

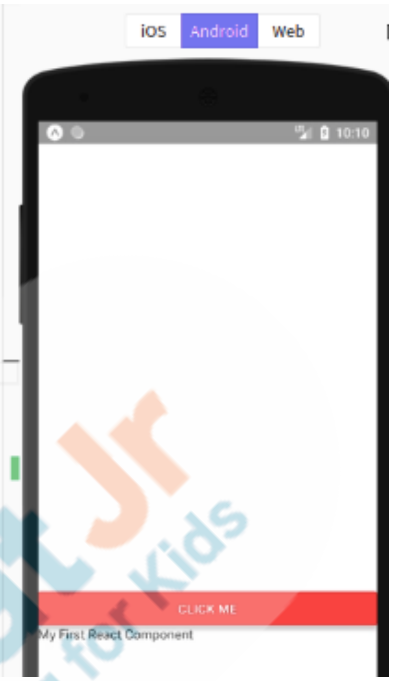
	We have created a Component class called RedButton. Now we can create the component by using the <RedButton/> inside the render function in App class.	The student uses the RedButton Component and renders it in the App.

<table>
<tr>
<td colspan="3">

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

class RedButton extends Component {
  render() {
    return <Button color="red" title="Click Me"/>;
  }
}

export default class App extends Component {
  render() {
    return (
      <View style={{ marginTop: 500 }}>
        <RedButton />
        <Text>My First React Component</Text>
      </View>
    );
  }
}
```



</td>
</tr>
<tr>
<td></td>
<td>Congrats, you have created your first React Native component.

We will be creating many more in the upcoming classes to create our apps.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>One more thing - A render function can return only one React Component. If there are more than one React Components that need to be returned, they should be nested inside View.

For example: If we want to render both Button and Text, then they should be enclosed inside View.</td>
<td>The student writes the code to display both - Text and Button.</td>
</tr>
</table>

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 5 Mins

Teacher starts slideshow **from slide 17 to slide 27**

Activity details	Solution/Guidelines
Run the presentation from slide 17 to slide 27 **Following are the wrap-up session deliverables:** • **Explain the facts and trivias** • **Next class challenge** • **Project for the day** • **Additional Activity**	Guide the student to develop the project and share with us.

Quiz time - Click on in-class quiz

Question	Answer
Which of the following is the correct component to display text?	**B**

A. <text></text> B. <Text></Text> C. <text /> D. <Text />	
Which of the following is true about the following statement export default class App extends Component{ } A. Component is the Baseclass and all React components are inherited from it and App is the child class of the Component. B. App is the Baseclass and all React components are inherited from it and the component is the child class of the App. C. Both Component and App are the Baseclasses. D. Both Component and App are the child classes.	**C**
Which of the following is true about javascript and HTML? A. HTML is a declarative language and Javascript is an imperative language. B. Javascript is a declarative language and HTML is an imperative language. C. Both are declarative languages. D. Both are imperative languages.	**A**
End the quiz panel	
FEEDBACK • **Encourage the student to play around with JSX.** • **Encourage the student to make reflection notes in the markdown format.** • **Complement the student for her/his effort in the class.**	
Right now, we have a button in our App which really does not do anything.	

<table>
<tr><td></td><td>In the next class, we will learn how to add functions to a component. We will perform an action when a Button is pressed - like playing a sound.

You get a “hats off”.

I hope you will look forward to the next class. Have a great day!</td><td>Make sure you have given at least 2 Hats Off during the class for:
 
 
  </td></tr>
<tr><td>Project Overview</td><td>Do you know this?

Goal of the Project:
Today, We have learnt two kinds of languages - imperative and declarative - and much more!

We will test your skills today. Lets Go!!

Story:

A good coder should always be ready to answer any questions related to programming. As you are a gaming coder now, let's see how much you remember!

I am very excited to see your project solution and I know you both will do really well.

Bye Bye!</td><td>Note: You can assign the project to the student in class itself by clicking on the Assign Project button which is available under the projects tab.

Students engage with the teacher over the project.</td></tr>
<tr><td colspan="3">Teacher ends slideshow  </td></tr>
</table>

Teacher Clicks	✖ End Class	
Additional Activities	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	Snack: My First App	https://snack.expo.io/@vishalgaddam873/my-first-app
Teacher Activity 2	Reference code	https://snack.expo.io/@vishalgaddam873/my-first-app:-teacher-reference
Student Activity 1	Expo Sign Up	https://expo.io
Student Activity 2	Snack: My First App	https://snack.expo.io/@vishalgaddam873/my-first-app

Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO_C53_withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/a2c77edb-aced-43b8-8261-3a6cf6c13a4b.pdf