




Topic	COMPLETING THE FACE RECOGNITION APP	
Class Description	Students will revisit the concepts of stack navigation in react native to complete the face recognition app.	
Class	C185	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> <li>Learn to add stack navigation in the app to create the first page in the App UI.</li> </ul>	
Resources Required	<ul style="list-style-type: none"> <li>Teacher Resources:               <ul style="list-style-type: none"> <li>Visual Studio Code Editor</li> <li>laptop with internet connectivity</li> <li>smartphone</li> <li>earphones with mic</li> <li>notebook and pen</li> </ul> </li> <li>Student Resources:               <ul style="list-style-type: none"> <li>Visual Studio Code Editor</li> <li>laptop with internet connectivity</li> <li>smartphone</li> <li>earphones with mic</li> <li>notebook and pen</li> </ul> </li> </ul>	
Class structure	<b>Warm-Up</b> <b>Teacher-led Activity</b> <b>Student-led Activity</b> <b>Wrap-Up</b>	<b>5 mins</b> <b>15 mins</b> <b>20 mins</b> <b>5 mins</b>
WARM-UP SESSION - 5 mins		
<u>CONTEXT</u> <ul style="list-style-type: none"> <li>Design the first page of the app's UI.</li> </ul>		

<div></div> <div><b>Teacher Starts Slideshow</b> <b>Slide 1 to 3</b> Refer to speaker notes and follow the instructions on each slide.</div>		
<div>Hey &lt;student's name&gt;. How are you? It's great to see you! Are you excited to learn something new today?</div> <div><b>Following are the WARM-UP session deliverables:</b><ul style="list-style-type: none"><li>Greet the student.</li><li>Revision of previous class activities.</li></ul></div>	<div><b>ESR:</b> Hi, thanks! Yes I am excited about it!</div> <div>Click on the slide show tab and present the slides</div>	
<div><b>WARM-UP QUIZ</b> Click on In-Class Quiz</div>		
<div></div> <div><b>Continue WARM-UP Session</b> <b>Slide 4 to 8</b></div>		
<div><b>Following are the session deliverables:</b><ul style="list-style-type: none"><li>Appreciate the student.</li><li>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</li></ul></div>		
<b>Class Steps</b>	<b>Teacher Action</b>	<b>Student Action</b>
<b>Step 1:</b> <b>Warm-Up</b> <b>(5 mins)</b>	Hi, how are you?  Great!	<b>ESR:</b> I am good!
	Can you tell me what we have learned in the previous class?	<b>ESR:</b> <ul style="list-style-type: none"><li>We learned how to create categories and then we created categories for our filters.</li></ul>

	<p>Great!</p> <p>Today, we will be completing our Face Recognition App. For this, we will add Stack Navigation to our code and then create a home screen for our App.</p> <p>Do you remember what stack navigation is?</p> <p>Great! So, are you excited for today's class?</p> <p>Let's get started then.</p>	<p><b>ESR:</b> It is a form of navigation available in react native to navigate between different screens. It stacks the screens on top of one another.</p> <p><b>ESR:</b> Yes.</p>
<p>Teacher Ends Slideshow </p>		
TEACHER-LED ACTIVITY - 15 mins		
Teacher Initiates Screen Share		
<p><b><u>CHALLENGE</u></b></p> <ul style="list-style-type: none"> <li>• Design App UI.</li> <li>• Adding Stack Navigation to the App.</li> </ul>		

<b>Step 2: Teacher-led Activity (15 mins)</b>	<p><u><a href="#">[Teacher Activity 1]</a></u></p> <p>To implement stack navigation in our App, we have to install certain modules in our app.</p> <p>Let's take the time to install them one by one so that we are ready to have our Stack Navigation integrated with our app -</p>	
<p><b><i>expo install react-native-gesture-handler</i></b></p> <p><b><i>expo install react-native-reanimated</i></b></p> <p><b><i>expo install react-native-screens</i></b></p> <p><b><i>expo install react-native-safe-area-context</i></b></p> <p><b><i>expo install @react-native-community/masked-view</i></b></p> <p>The things we just installed are libraries that the navigation uses. All these modules are also mentioned in the documentation for stack navigation (refer to Teacher Activity 1 / Student Activity 1), and now we can install our navigation -</p> <p><b><i>yarn add @react-navigation/native</i></b></p> <p><b><i>yarn add @react-navigation/stack</i></b></p>		
	<p>Okay, now the first thing that we want to do is, we will create a <b>StackNavigator</b> and then create our navigation structure in our <b>App.js</b> file -</p>	

### App.js

```
import 'react-native-gesture-handler';
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';

import Home from "../screens/Home";
import Main from "../screens/Main";

const Stack = createStackNavigator();

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home" screenOptions={{
        headerShown: false
      }}>
        <Stack.Screen name="Home" component={Home} />
        <Stack.Screen name="Main" component={Main} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

Here, we are first importing **NavigationContainer & createStackNavigator**.

**createStackNavigator** helps us create a navigator where we can define our Screens and these screens should always be inside a Navigation Container.

When we were in the React Native module, we used navigation like Drawer Navigation and Tab Navigation. All these navigations require their screens to be inside a **<NavigationContainer>** component.

We had our **<Main>** component already completed in the previous class, which is our screen with the camera where the user can try out different frames, but we have also imported a new screen here called **<Home>**. This is what we will be building up today.

For now, let's create a file called **Home.js** inside our **screens** folder and give the following boilerplate code to it.



**./screens/Home.js**

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

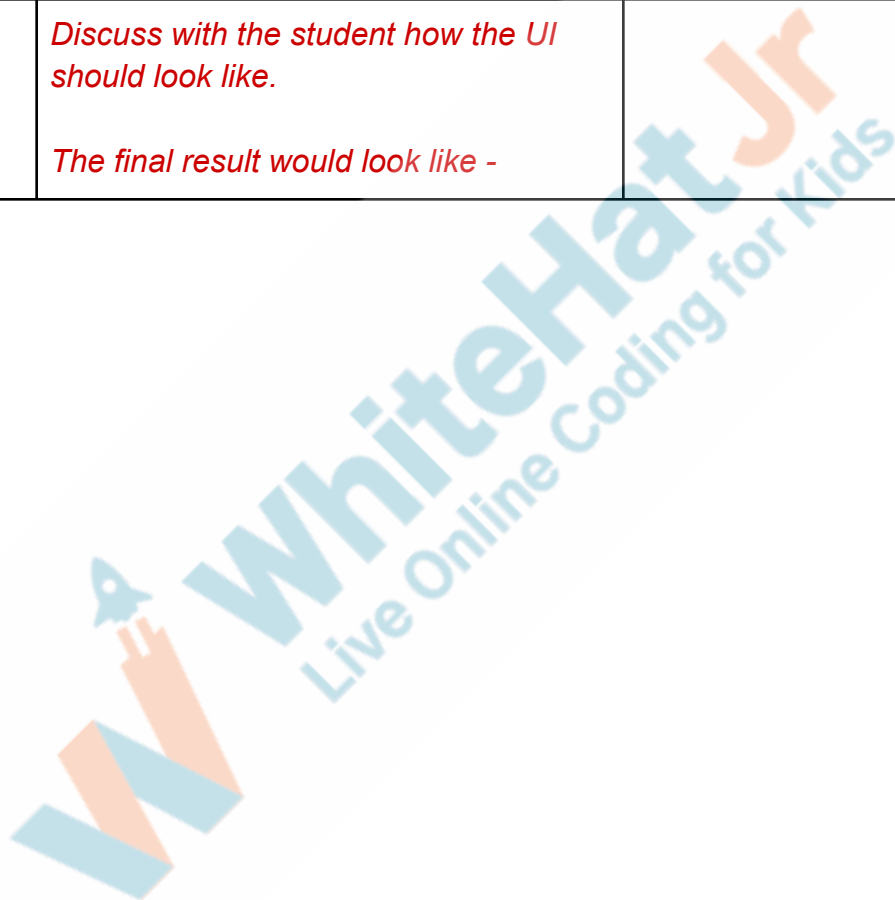
export default class Home extends Component {
  render() {
    return (
      <View
        style={{
          flex: 1,
          justifyContent: "center",
          alignItems: "center"
        }}>
        <Text>Home Screen!</Text>
      </View>
    )
  }
}
```

Awesome! If we run the app now, we can see just **"Home Screen!"** written on the screen.

That's because we specified in our Stack Navigator that we want our default screen (or the `initialRouteName`) to be Home.

	As you might have guessed already, now it's going to be your turn to complete this app by building the home screen and adding the final navigation from Home screen to the Main Screen.	
<b>Teacher Stops Screen Share</b>		
	Now it's your turn. Please share your screen with me.	
<div>  <p><b>Teacher Starts Slideshow</b> <b>Slide 9 to 11</b> Refer to speaker notes and follow the instructions on each slide.</p> </div>		
<p>We have one more class challenge for you. Can you solve it?</p> <p>Let's try. I will guide you through it.</p>		
<div>  <p><b>Teacher Ends Slideshow</b></p> </div>		
<b>STUDENT-LED ACTIVITY - 20 mins</b>		
<ul style="list-style-type: none"> <li>• Ask the student to press the <b>ESC</b> key to come back to the panel.</li> <li>• Guide the student to start screen share.</li> <li>• Teacher gets into fullscreen.</li> </ul>		
<p style="text-align: center;"><b><u>ACTIVITY</u></b></p> <ul style="list-style-type: none"> <li>• Design the app UI.</li> <li>• Create the final screen and complete the App.</li> </ul>		

<b>Step 3: Student-led Activity (20 mins)</b>	<p><u><a href="#">[Student Activity 1]</a></u></p> <p><b>Note:</b> The student will repeat teacher activity for adding stack navigation.</p> <p>Guide the student to achieve the same and share the code snippets.</p>	
	<p>Discuss with the student how the UI should look like.</p> <p>The final result would look like -</p>	







*Guide the student to create the header of the screen with App Name and App's Catch Phrase.*

```
<View style={styles.container}>
  <SafeAreaView style={styles.droidSafeArea} />
  <View style={styles.headingContainer}>
    <View style={{ flexDirection: 'row', flexWrap: 'wrap' }}>
      <Text style={styles.titleText1}>FR</Text><Text
style={styles.titleText2}>APP</Text>
    </View>
    <View style={{ flexDirection: 'row', flexWrap: 'wrap' }}>
      <Text style={styles.subheading1}>Try Out</Text><Text
style={styles.subheading2}> Cool Frames</Text>
    </View>
  </View>
</View>
```

```
container: {
  flex: 1,
  backgroundColor: "#6278e4"
},
droidSafeArea: {
  marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 0
},
headingContainer: {
  flex: 0.2,
  alignItems: 'center',
  justifyContent: 'center'
},
titleText1: {
  fontSize: RFValue(30),
  fontWeight: "bold",
  color: "#efb141",
  fontStyle: 'italic',
  textShadowColor: 'rgba(0, 0, 0, 0.75)',
  textShadowOffset: { width: -3, height: 3 },
  textShadowRadius: 1
},
titleText2: {
```

```

fontSize: RFValue(30),
fontWeight: "bold",
color: "white",
fontStyle: 'italic',
textShadowColor: 'rgba(0, 0, 0, 0.75)',
textShadowOffset: { width: -3, height: 3 },
textShadowRadius: 1
},
subheading1: {
  fontSize: RFValue(20),
  color: "#efb141",
  fontStyle: 'italic',
  textShadowColor: 'rgba(0, 0, 0, 0.75)',
  textShadowOffset: { width: -3, height: 3 },
  textShadowRadius: 1
},
subheading2: {
  fontSize: RFValue(20),
  color: "white",
  fontStyle: 'italic',
  textShadowColor: 'rgba(0, 0, 0, 0.75)',
  textShadowOffset: { width: -3, height: 3 },
  textShadowRadius: 1
},

```

*Guide the student to create the second section for text and images -*

```

<View style={styles.contentContainer}>
  <View style={{ flex: 0.5 }}>
    <Text style={styles.contentText}>Experience the virtual experience of
    trying out different frames from our wide collection on your mobile phone before
    making any purchase, just how you would in an offline store!</Text>
  </View>
  <View style={{ flexDirection: "row", flex: 0.25 }}>

```

```

        <View style={{ flex: 0.5 }}>
            <Image source={require('../assets/Frapp-03.png')} style={{ height:
64, width: 160 }} />
        </View>
        <View style={{ flex: 0.5 }}>
            <Image source={require('../assets/Frapp-09.png')} style={{ height:
64, width: 160 }} />
        </View>
    </View>
    <View style={{ flexDirection: "row", flex: 0.25 }}>
        <View style={{ flex: 0.5 }}>
            <Image source={require('../assets/Frapp-02.png')} style={{ height:
64, width: 160 }} />
        </View>
        <View style={{ flex: 0.5 }}>
            <Image source={require('../assets/Frapp-08.png')} style={{ height:
64, width: 160 }} />
        </View>
    </View>
</View>

```

```

contentContainer: {
    flex: 0.6,
    margin: RFValue(5),
    borderRadius: RFValue(15),
    backgroundColor: "white",
    height: "100%",
    padding: RFValue(20)
},
contentText: {
    fontSize: RFValue(17),
    fontStyle: 'italic',
    fontWeight: "bold"
},



```

*Guide the student to create the final section with the button -*

```
<View style={styles.buttonContainer}>
  <TouchableOpacity onPress={() =>
this.props.navigation.navigate("Main")}>
    <View style={styles.button}>
      <Text style={styles.buttonText}>Try Now!</Text>
    </View>
  </TouchableOpacity>
</View>
```

```
buttonContainer: {
  flex: 0.2,
  justifyContent: "center",
  alignItems: "center"
},
button: {
  backgroundColor: "#efb141",
  paddingLeft: RFValue(50),
  paddingRight: RFValue(50),
  paddingTop: RFValue(20),
  paddingBottom: RFValue(20),
  borderRadius: RFValue(20)
},
buttonText: {
  fontSize: RFValue(25),
  fontStyle: 'italic',
  color: "white",
  textShadowColor: 'rgba(0, 0, 0, 0.75)',
  textShadowOffset: { width: -1, height: 1 },
  textShadowRadius: 1
}
```

Don't forget to add the navigation on the **onPress()** event of our **<TouchableOpacity/>**.

	<i>Guide the student to test the output.</i>	
<b>Teacher Guides Student to Stop Screen Share</b>		
<b>WRAP UP SESSION - 5 mins</b>		
<div> <div>Teacher Starts Slideshow</div> <div>Slide 12 to 16</div>  </div>		
<b>Activity details</b> <b>Following are the WRAP-UP session deliverables:</b> <ul style="list-style-type: none"> <li>• Appreciate the student.</li> <li>• Revise the current class activities.</li> <li>• Discuss the quizzes.</li> </ul>		
<b>WRAP-UP QUIZ</b> Click on In-Class Quiz		
<div> <div>Continue WRAP-UP Session</div> <div>Slide 17 to 22</div>  </div>		
<b>Activity Details</b> <b>Following are the session deliverables:</b> <ul style="list-style-type: none"> <li>• Explain the facts and trivia</li> <li>• Next class challenge</li> <li>• Project for the day</li> <li>• Additional Activity (Optional)</li> </ul>		
<p align="center"><b><u>FEEDBACK</u></b></p> <ul style="list-style-type: none"> <li>• <b>Appreciate and compliment the student for trying to learn a difficult concept.</b></li> <li>• <b>Get to know how they are feeling after the session.</b></li> <li>• <b>Review and check their understanding.</b></li> </ul>		
<b>Teacher Action</b>		<b>Student Action</b>

<p>You get Hats off for your excellent work!</p>		<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div> <div>Creatively Solved Activities +10</div> <div>Great Question +10</div> <div>Strong Concentration +10</div> </div>
<p><b>PROJECT OVERVIEW DISCUSSION</b> Refer the document below in Activity Links Sections</p>		
<p><b>Teacher Clicks</b></p>		<p>✕ End Class</p>
<p><b>Additional Activities</b></p>	<p><i>Encourage the student to write reflection notes in their reflection journal using markdown.</i></p> <p>Use these as guiding questions:</p> <ul style="list-style-type: none"> <li>• What happened today? <ul style="list-style-type: none"> <li>○ Describe what happened.</li> <li>○ The code I wrote.</li> </ul> </li> <li>• How did I feel after the class?</li> <li>• What have I learned about programming and developing games?</li> <li>• What aspects of the class helped me? What did I find difficult?</li> </ul>	<p><i>The student uses the markdown editor to write their reflections in a reflection journal.</i></p>

Activity	Activity Name	Links
Teacher Activity 1	Previous Class Code	<a href="https://github.com/whitehatjr/PRO-C184-Code-Ref">https://github.com/whitehatjr/PRO-C184-Code-Ref</a>
Teacher Activity 2	Stack Navigation	<a href="https://reactnavigation.org/docs/getting-started">https://reactnavigation.org/docs/getting-started</a>
Teacher Activity 3	Final Reference Code	<a href="https://github.com/whitehatjr/PRO-C185-Code-Ref">https://github.com/whitehatjr/PRO-C185-Code-Ref</a>
Student Activity 1	Previous Class Code	<a href="https://github.com/whitehatjr/PRO-C184-Code-Ref">https://github.com/whitehatjr/PRO-C184-Code-Ref</a>
Student Activity 2	Stack Navigation	<a href="https://reactnavigation.org/docs/getting-started">https://reactnavigation.org/docs/getting-started</a>
Teacher Reference 1	Project Document	<a href="https://s3-whjr-curriculum-uploads.whjr.online/82a7fd58-8199-4818-90e9-f61e87a1c071.pdf">https://s3-whjr-curriculum-uploads.whjr.online/82a7fd58-8199-4818-90e9-f61e87a1c071.pdf</a>
Teacher Reference 2	Project Solution	<a href="https://github.com/whitehatjr/AR-PRO-C185">https://github.com/whitehatjr/AR-PRO-C185</a>
Teacher Reference 3	Visual-Aid	<a href="https://s3-whjr-curriculum-uploads.whjr.online/f81e919f-a67c-4c5c-a902-b4f7b5820979.html">https://s3-whjr-curriculum-uploads.whjr.online/f81e919f-a67c-4c5c-a902-b4f7b5820979.html</a>
Teacher Reference 4	In-Class Quiz	<a href="https://s3-whjr-curriculum-uploads.whjr.online/67a51209-6034-44d2-9ef6-fc033d21f76d.pdf">https://s3-whjr-curriculum-uploads.whjr.online/67a51209-6034-44d2-9ef6-fc033d21f76d.pdf</a>