Topic	e-library APP	
Class Description	**Students build a simple and innovative library management system for a low income school that cannot afford a librarian.** **Students think about the problem and design a wireframe for the application. Students also learn about top and bottom tab navigation which they can use in their app.**	
Class	**C68**	
Class time	**60 mins**	
Goal	● Explore a case study of a wireless library management system for a low income school. ● Design a wireframe for the wireless library application. ● Build tab navigation to issue/return and search for a book.	
Resources Required	● Teacher Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ VS Code Editor ○ Android/iOS Smartphone with Expo App installed ● Student Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ VS Code Editor ○ Android/iOS Smartphone with Expo App installed	
Class structure	**Warm-Up** **Teacher-led Activity** **Student-led Activity** **Wrap-Up**	**10 mins** **20 min** **25 min** **5 min**
Credit:	MaterialBottomTab **Navigator documentation is**	

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	licensed under MIT license.	

WARM-UP SESSION - 10 mins	
Teacher starts slideshow from slides 1 to 13 Refer to speaker notes and follow the instructions on each slide.	
Activity details	**Solution/Guidelines**
Hi, how have you been? Are you excited to learn something new? **Run the presentation from slide 1 to slide 4.** **The following are the warm-up session deliverables:** ● Reconnect with previous class topics. ● Warm-Up quiz session.	**ESR**: Varied Response. Click on the slide show tab and present the slides.
QnA Session	
Question	**Answer**
What does git push do? A. pushes the remote to the current local repository B. pushes the current local repository to remote C. clones the remote repository into a local repository D. deletes the repository	**B**
What command adds the work to the HISTORY of the works you are doing? A. Git add B. Git clone C. Git remote add D. Git commit	**D**

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Continue the warm-up session	
Activity details	**Solution/Guidelines**
Run the presentation from slide 5 to slide 13 to set the problem statement. **The following are the warm-up session deliverables:** ● Explore case study for a library management system app for a low income school which cannot afford a dedicated librarian.	Narrate the story by using hand gestures and voice modulation methods to bring in more student interest.
Teacher ends slideshow	
TEACHER-LED ACTIVITY - 20 mins	
Teacher Initiates Screen Share	
CHALLENGE ● **Design a wireframe for the wireless library management application.**	
Teacher Action	**Student Action**
In this and the coming few classes, we will be working on the problems of a low income school which cannot afford a dedicated librarian. In the process, we will create an application which will help solve this problem for National Public School. Let's get started.	
So, what kind of an app would help the school with this problem? Guide the student to think about the different app features:	*The student points out details about the app features.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

• issue of book; • return of book; • search for book's availability; and • search for books issued to a particular student.	
To help think clearly about what an app will do and what features it will have for its users, developers and product engineers often write detailed user stories. Each user story is a small, independently contained feature in an app which fulfills a user need. An app will have several user stories. A good user story goes like this: As a (<who is going to use the app>), I want to (<what action is the user going to take through the app>), so that (<what benefit will the user get out of the app>). If you observe, the user story answers the **WHO, WHAT** and **WHY** for an app feature you are going to build. This helps the developers and everyone involved in building the app bring more clarity as to what the app will do. For example: One user story in our app will be: As a teacher, I want to enter the student id, book id and submit the info so that I can issue the book to any student. Can you write all the different user stories for this particular app we are trying to build here?	*The student listens to the sample user story.* *Student spend the next 5-7 mins writing all the user stories for the app.*
Teacher checks for the student's understanding. User stories for Teacher reference:	*The student talks about each user story in depth.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

• As a teacher, I want to enter the student id, book id and submit the info so that I can issue the book to any student. • As a teacher, I want to enter the student id, book id and submit the info so that I can return the book issued by the student back to the library. • As a teacher, I want to be able to search for any book's availability in the library so that I can inform the students about the availability of any book. • As a teacher, I want to search for all the books issued by a particular student so that I can check for the books that were issued by the student. *Note: The student might have come up with more user stories.* *Dissect each user story and ask the student to explain each of their user stories.*	
Let's say all the teachers of NPS have this app installed on their phones. Now what will be the user flow of our app for a teacher in National Public School? *Teacher can fill in the gaps.*	**ESR:** • Any teacher who is free in a particular period can use the app to issue or return a book. • Each book will have a unique id on it. Each student will also have a unique id. • The teacher can enter this information

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

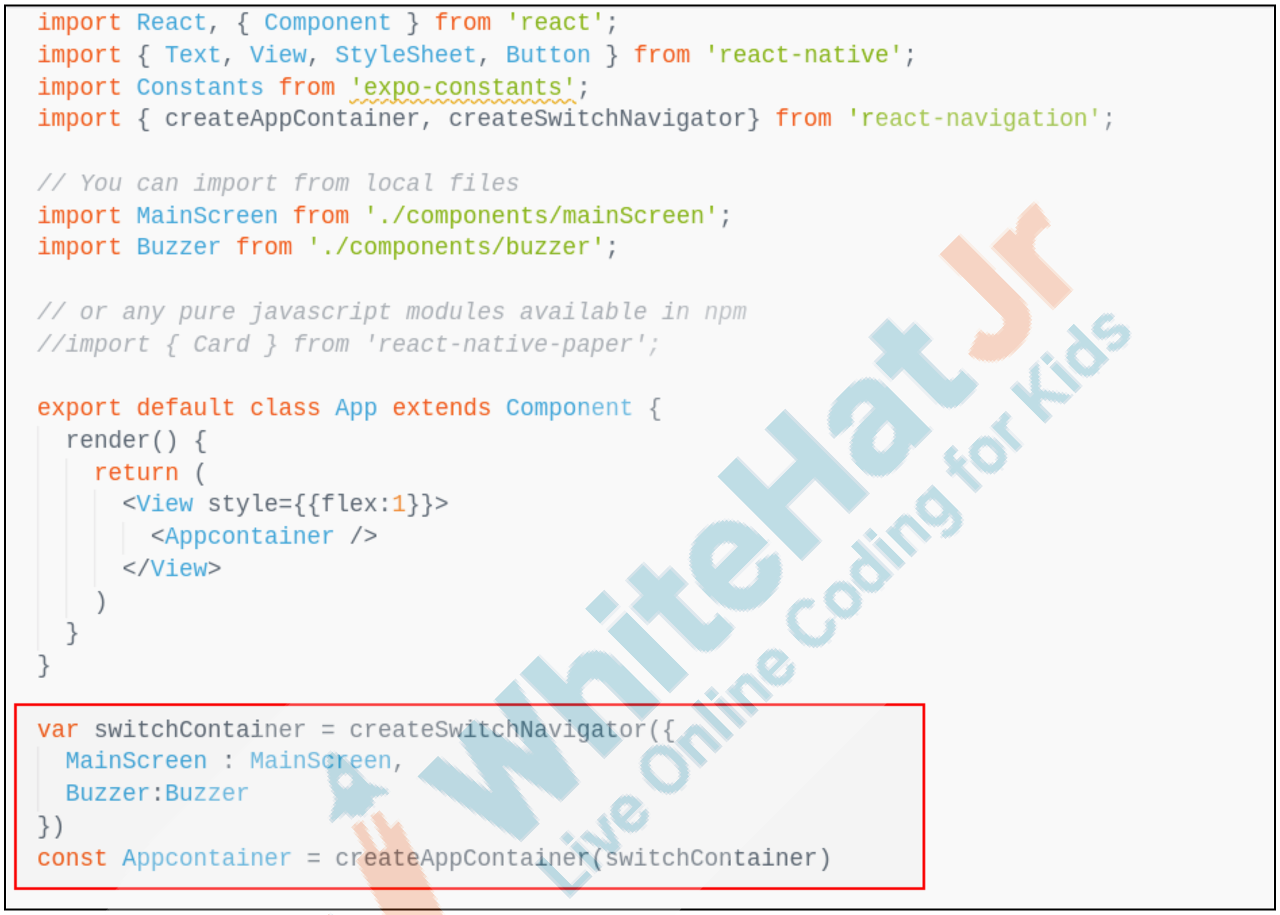
	(unique book id and student id) to issue the book to the student and store the information as an issued book. ● If the student has come up to return a book, the teacher will enter the same information (unique book id and student id) to return the book back to the library. ● The teacher can also search for a book by its id to check its availability in the library. ● The teacher can also search for books issued by any one particular student.
Awesome. Based on our current understanding, the teacher can either issue/return (both ask for the same information) or search for any book or books issued by a student. Can you quickly draw a wireframe of our application? Also, let's give our application a name - let's call it **e-library.**	*The student draws the wireframe of the app.* *The wireframe should have two different screens:*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Allow the student some time to draw the wireframe of the app.	*Screen1: Issue/Return a book* *Screen2: Search for book*
Give compliments to the student on the design of their wireframe. Our app has two screens. Do you remember what we did last time when we had to move between the two screens? How did we create the navigation? *Guide the student to review the QuizBuzzer App where they created switch navigation from Student Activity 1.*	**ESR:** We switched between two screens by pressing a button. ESR: We used the **SwitchNavigator**. *Student opens the link from Student Activity 1* *Student reviews the SwitchNavigator.*
Can you explain how we created the Switch navigation in our app?	ESR: ● We created two screens - Buzzer Screen and main screen. ● In app.js we created a switch Navigator using create Switch Navigator. It contained two screens. ● We created an AppContainer component using **createAppContainer()**.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	• We then used the AppContainer in our application.



```
import React, { Component } from 'react';
import { Text, View, StyleSheet, Button } from 'react-native';
import Constants from 'expo-constants';
import { createAppContainer, createSwitchNavigator} from 'react-navigation';

// You can import from local files
import MainScreen from './components/mainScreen';
import Buzzer from './components/buzzer';

// or any pure javascript modules available in npm
//import { Card } from 'react-native-paper';

export default class App extends Component {
  render() {
    return (
      <View style={{flex:1}}>
        <Appcontainer />
      </View>
    )
  }
}

var switchContainer = createSwitchNavigator({
  MainScreen : MainScreen,
  Buzzer:Buzzer
})
const Appcontainer = createAppContainer(switchContainer)
```

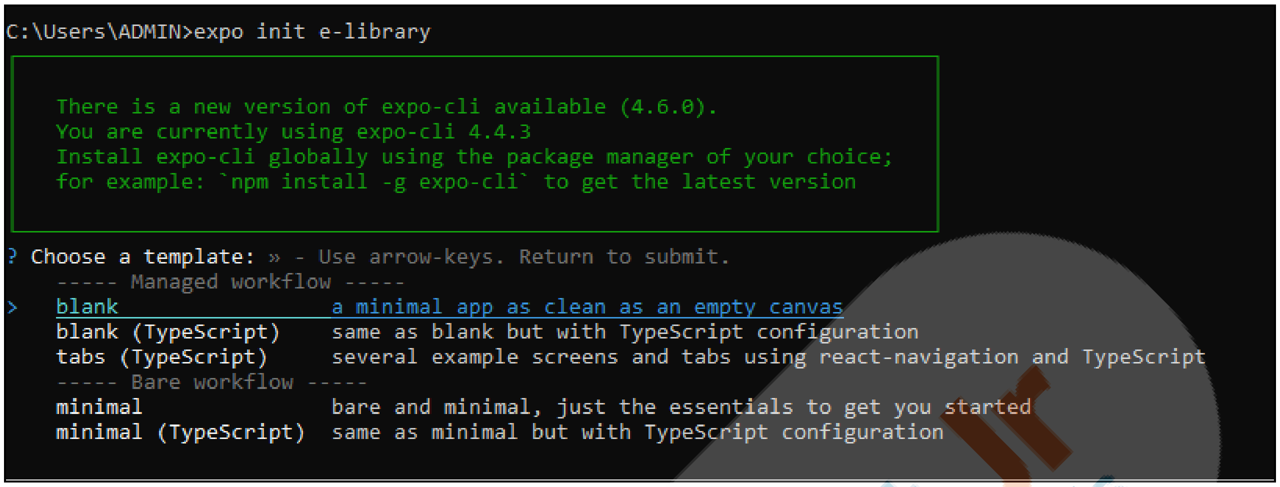
Great! Creating a Tab Navigator is exactly the same. Let's start our app by creating a simple Tab Navigator which can help us switch between the two screens - issue/return and search. Also, we will be completely coding on our local machine this time. Let's get started.	

Teacher Stops Screen Share

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Now it's your turn. Please share your screen with me.	
STUDENT-LED ACTIVITY - 25 mins		
● **Ask the student to press the ESC key to come back to the panel.** ● **Guide the student to start Screen Share.** ● **The teacher gets into Fullscreen.**		
ACTIVITY ● **Build tab navigation to issue/return or search for a book**		
Teacher starts slideshow : Slide 14 to Slide 15		
Run the presentation slide 14 and 15 to set the student activity context.		
Teacher ends slideshow		
Note:- This is the very beginning of the app, as we are starting the app from scratch we are not providing any boilerplate code for the same. Let's start a new expo project. Guide the student to start the new expo project. **Note: Make sure that you are using the latest version of expo cli.**		*The student uses the expo init command to create a new expo project.* *He/she chooses a blank template, gives a slug name to the app and waits for all the expo package installations to complete.*

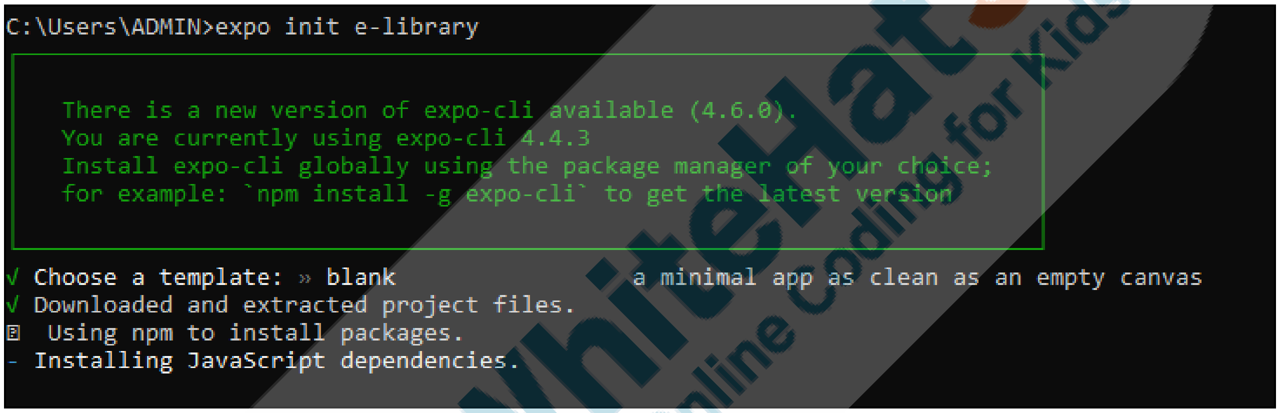
© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



```
C:\Users\ADMIN>expo init e-library

   There is a new version of expo-cli available (4.6.0).
   You are currently using expo-cli 4.4.3
   Install expo-cli globally using the package manager of your choice;
   for example: `npm install -g expo-cli` to get the latest version

? Choose a template: » - Use arrow-keys. Return to submit.
    ----- Managed workflow -----
>   blank                 a minimal app as clean as an empty canvas
    blank (TypeScript)    same as blank but with TypeScript configuration
    tabs (TypeScript)     several example screens and tabs using react-navigation and TypeScript
    ----- Bare workflow -----
    minimal               bare and minimal, just the essentials to get you started
    minimal (TypeScript)  same as minimal but with TypeScript configuration
```



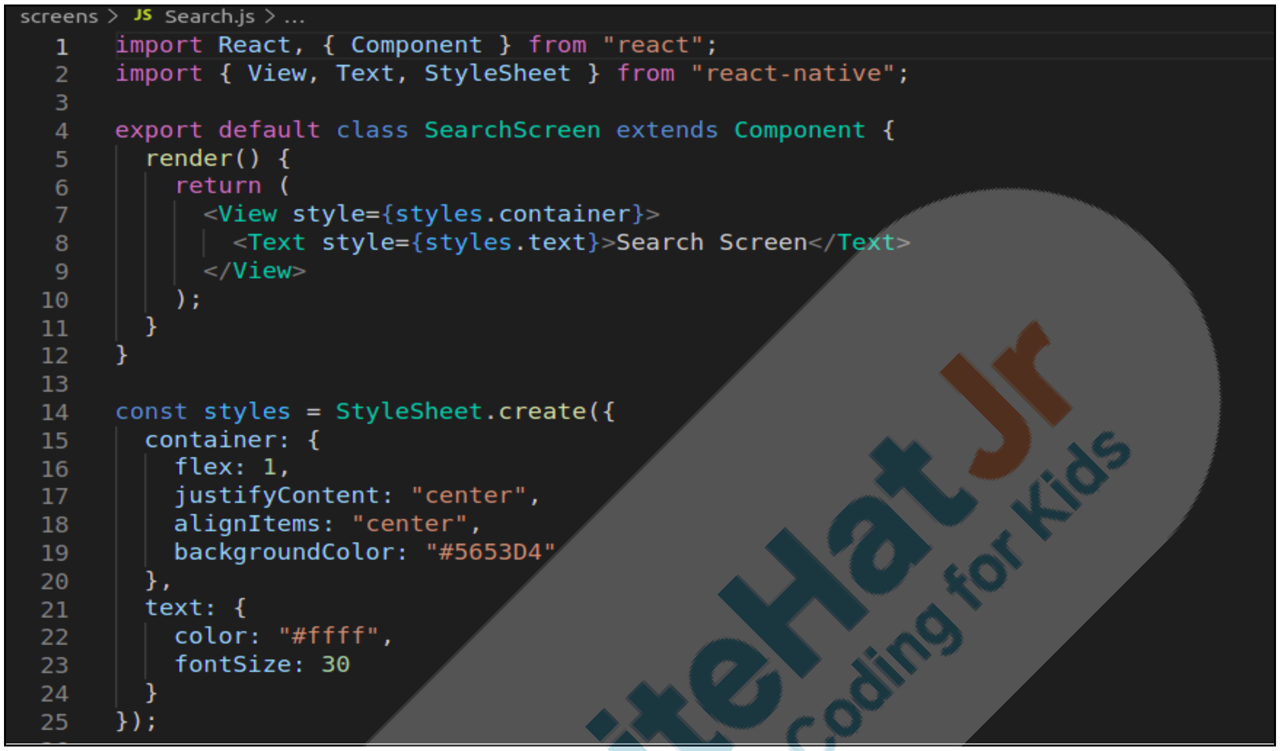
```
C:\Users\ADMIN>expo init e-library

   There is a new version of expo-cli available (4.6.0).
   You are currently using expo-cli 4.4.3
   Install expo-cli globally using the package manager of your choice;
   for example: `npm install -g expo-cli` to get the latest version

√ Choose a template: » blank                 a minimal app as clean as an empty canvas
√ Downloaded and extracted project files.
⊡  Using npm to install packages.
- Installing JavaScript dependencies.
```

This app will be a blank project. Let's create a screens folder in our app directory called - screens. Let's create two files **Search.js** and **Transaction.js** inside the screens folder. Inside these files let's create two components called **SearchScreen** and **TransactionScreen** and remember to import all the necessary packages to create the component. Also, remember to export the component so that we can use it in another screen.	*The student creates two screen components - SearchScreen and TransactionScreen in separate js files and exports them.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

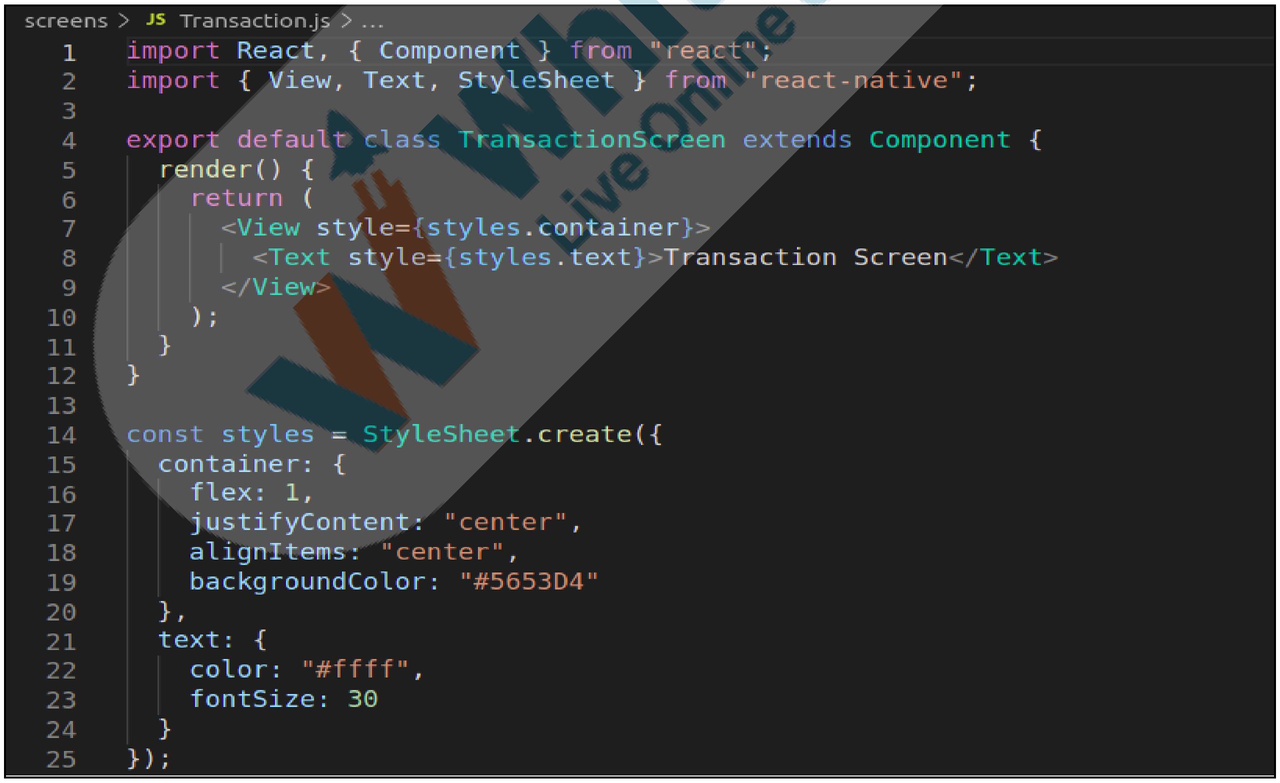


screens > JS Search.js > ...

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

export default class SearchScreen extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.text}>Search Screen</Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "#5653D4"
  },
  text: {
    color: "#ffff",
    fontSize: 30
  }
});
```



screens > JS Transaction.js > ...

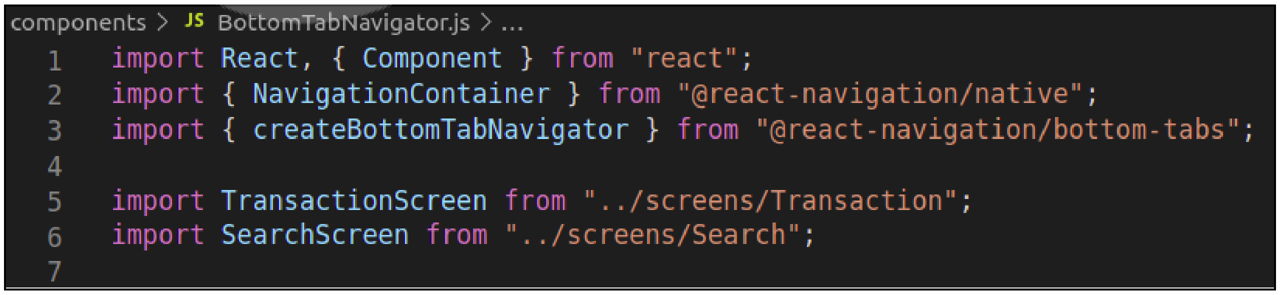
```
import React, { Component } from "react";
import { View, Text, StyleSheet } from "react-native";

export default class TransactionScreen extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.text}>Transaction Screen</Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "#5653D4"
  },
  text: {
    color: "#ffff",
    fontSize: 30
  }
});
```

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Let's install the Bottom Tab navigator to our app. If you are using npm to install the packages, use **npm install @react-navigation/bottom-tabs** command. Or if you are using yarn to install packages, use **yarn add @react-navigation/bottom-tabs** And also install bottom tab navigation peer dependencies **npm install react-native-screens** and **npm install react-native-gesture-handler**. Plus, install react navigation to create a navigation container using **npm install @react-navigation/native**.	
Now let's create a new file called BottomTabNavigator.js file. We'll create a folder called components and inside that folder we'll create the BottomTabNavigator.js file. Let's import both the screens and all the necessary react, react-native packages. Also, **import {NavigationContainer} from '@react-navigation/native'** package and **import {createBottomTabNavigator} from '@react-navigation/bottom-tabs'.** createBottomTabNavigator will help us create the bottom tab navigator.	*The student imports the packages in the BottomTabNavigator.js file.*


components > JS BottomTabNavigator.js > ...

```
import React, { Component } from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";

import TransactionScreen from "../screens/Transaction";
import SearchScreen from "../screens/Search";

```

WhiteHat Jr Live online Coding for Kids

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

WhiteHat Jr
Live online Coding for Kids

Here we'll create a Navigation container and inside it we'll create a tab navigator where we'll add the screens which we created.

First we'll create a variable called **Tab.**

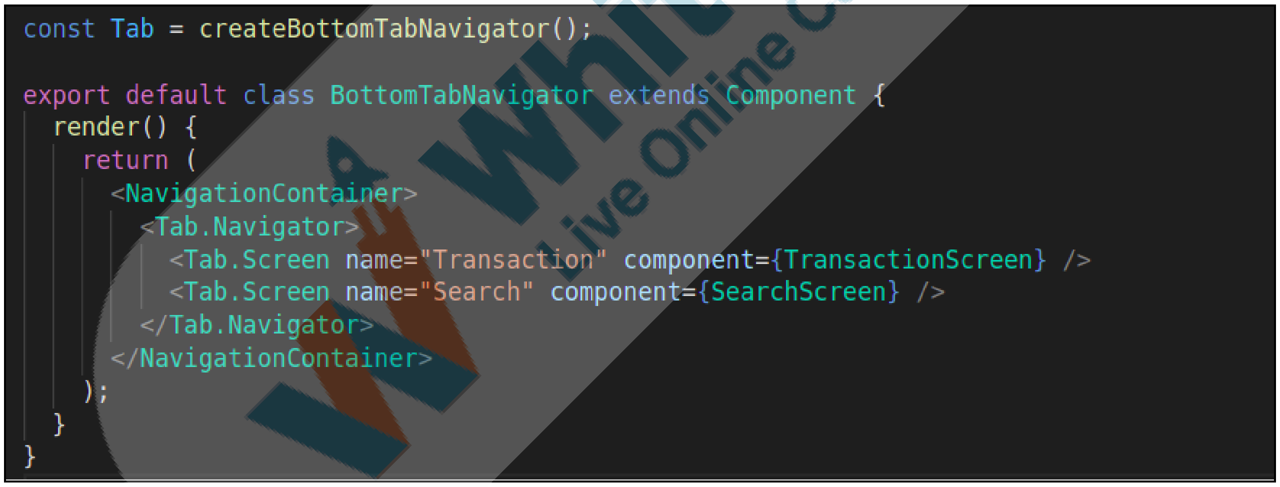
Inside this we'll call the createBottomTabNavigator function.

In the return function we'll create the Navigation container.

Inside the container we'll add the Tab.Navigator component.

Inside this component we'll add both the screens that we created earlier.

The student creates the Tab navigator component and passes the screens inside it.



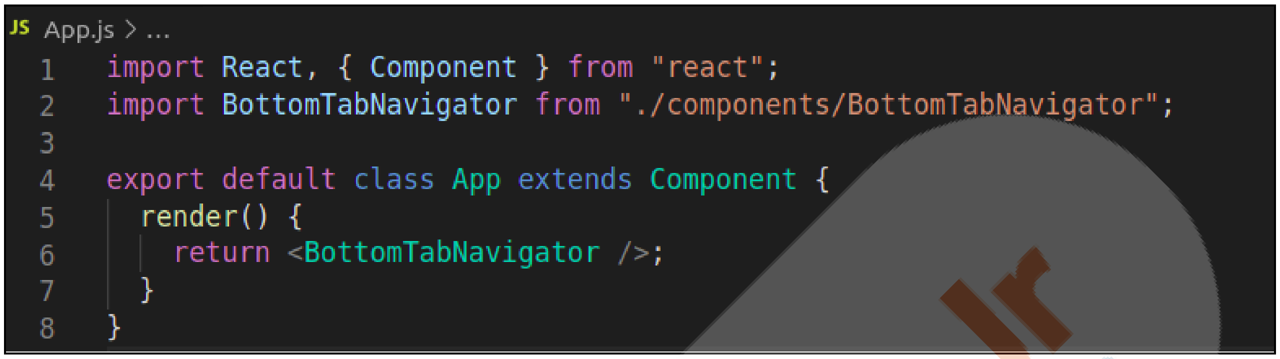
```
const Tab = createBottomTabNavigator();

export default class BottomTabNavigator extends Component {
  render() {
    return (
      <NavigationContainer>
        <Tab.Navigator>
          <Tab.Screen name="Transaction" component={TransactionScreen} />
          <Tab.Screen name="Search" component={SearchScreen} />
        </Tab.Navigator>
      </NavigationContainer>
    );
  }
}
```

We have our BottomTabNavigator ready now.
Now we just need to pass the BottomTabNavigator in the App.js file as the App.js file acts as the main entry point for our app.

The student imports the BottomTabNavigator and then passes it in the return function.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

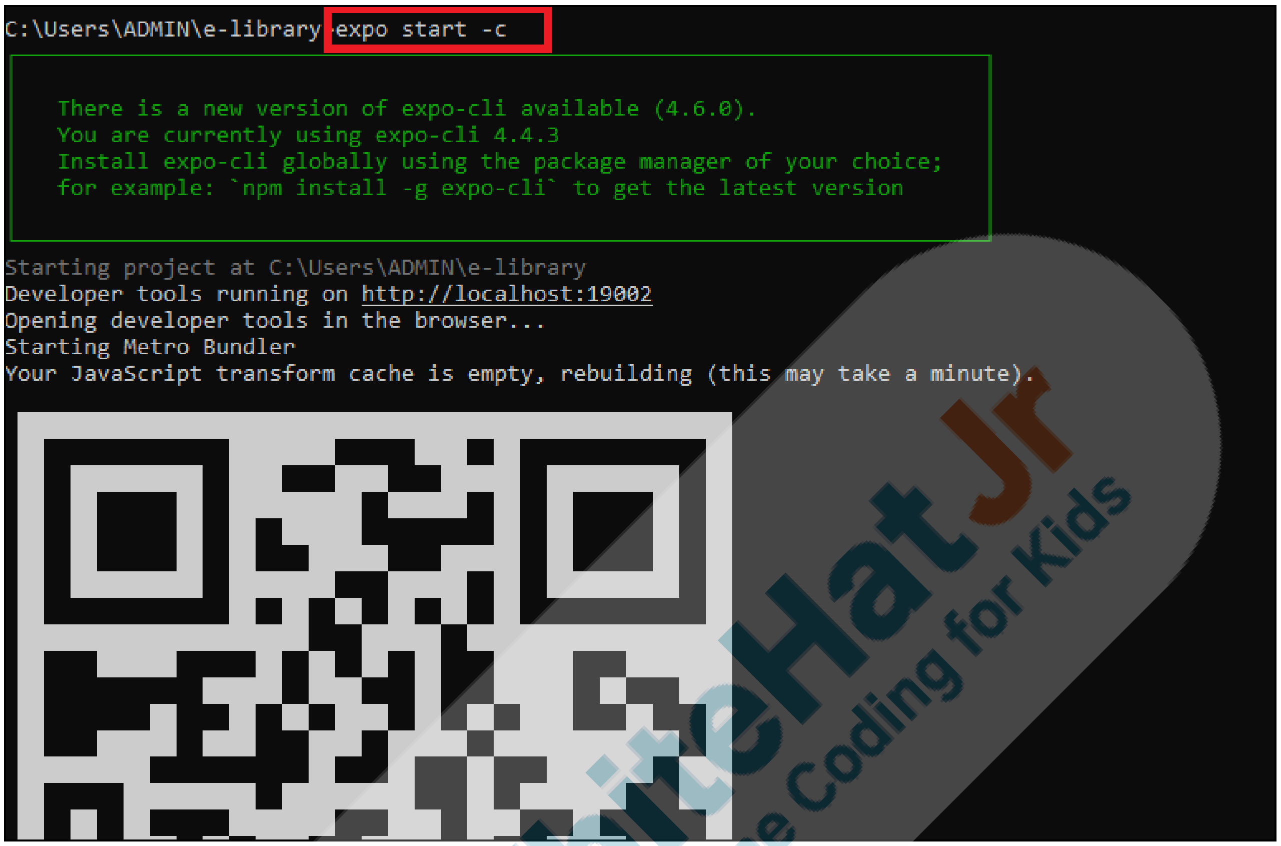
So we'll just import the BottomTabNavigator in the App.js file and then pass this component in the return function.	
	
Awesome job! Let's try to run the code. Use **expo start** to run the code.	*The student runs the code using expo start and checks the output on their phone by scanning the QR code.*

App.js > ...

```
import React, { Component } from "react";
import BottomTabNavigator from "./components/BottomTabNavigator";

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

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



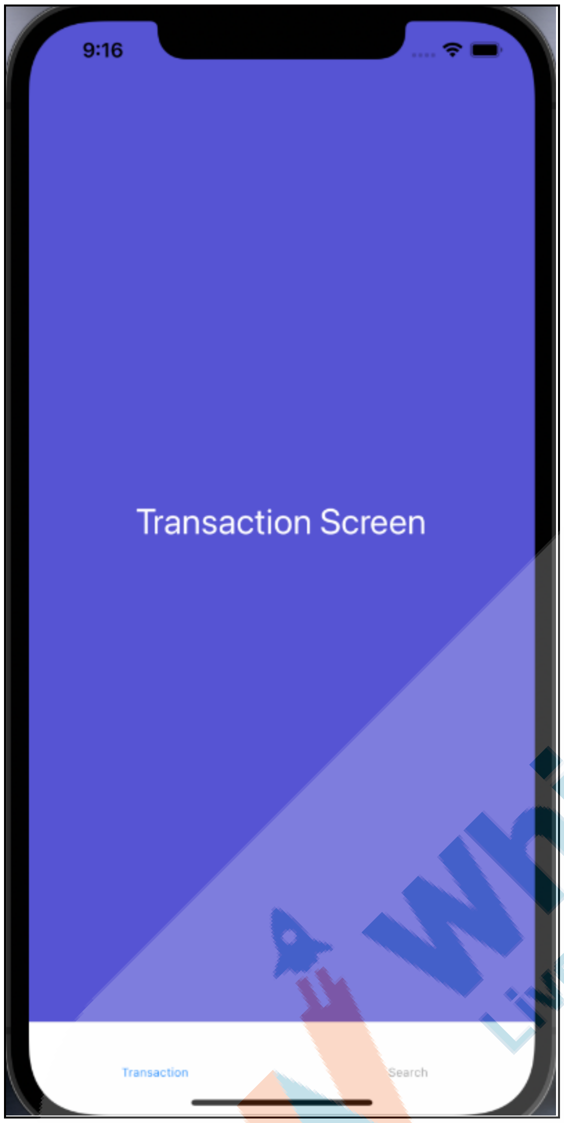
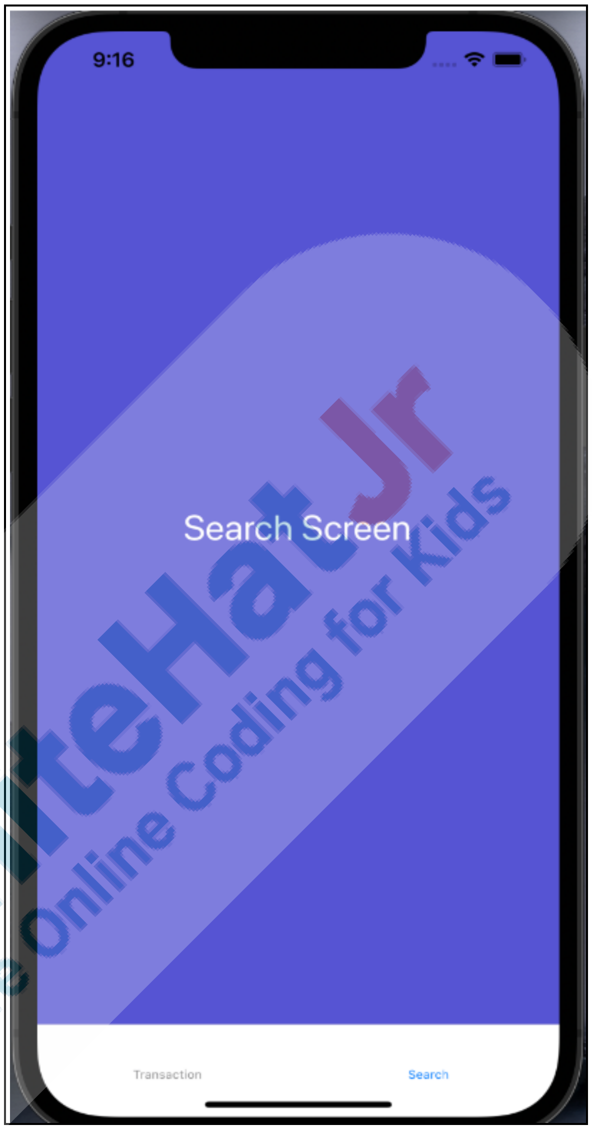
```
C:\Users\ADMIN\e-library>expo start -c

    There is a new version of expo-cli available (4.6.0).
    You are currently using expo-cli 4.4.3
    Install expo-cli globally using the package manager of your choice;
    for example: `npm install -g expo-cli` to get the latest version

Starting project at C:\Users\ADMIN\e-library
Developer tools running on http://localhost:19002
Opening developer tools in the browser...
Starting Metro Bundler
Your JavaScript transform cache is empty, rebuilding (this may take a minute).
```

What do you see?	**ESR:** I see two tabs which can be tapped to switch between the two screens.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

 	
Amazing! React Navigation has more documentation about how to use Tab Navigation. You can take a look at it in Student Activity 2 and Student Activity 3. For example: You can also experiment with **createMaterialTopTabnavigator** and	*The student experiments with createMaterialTopTabnavigator and createMaterialBottomTabNavigator to check the change in output.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

createMaterialBottomTabNavigator in place of createBottomTabNavigator.	
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 5 Mins	
Teacher starts slideshow from slide 16 to slide 26	
Activity details	**Solution/Guidelines**
Run the presentation from slide 16 to slide 26 **Following are the warm 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**
createBottomTabNavigator is present in which library? A. react-native B. react-navigation C. react-navigation-tabs D. react	**C**
What does command **expo start** do? A. expo start creates new app B. expo start creates a play store ready app C. expo start runs the app on localhost server D. expo start installs the node modules	**C**

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

A good User Story: A. helps developers think clearly about what an app will do. B. is a small, independently contained feature in an app which fulfills a user's need. C. can be changed or rewritten right up until the development stage. D. works as a skeleton of an app to clearly understand the structure of the app.	**D**
End the quiz panel	
Can you see how built-in components help us add features in our app very quickly? React-Navigation is a built-in package which most developers use when they build navigation in their apps. How are you feeling after the class?	**ESR:** Varied. **ESR:** Varied.
In the next classes, we will learn to scan the QR-Code and display the data inside the Text input.	
FEEDBACK ● **Guide the student to explore more tab navigation documentation and look at other features available.**	
Teacher Action	**Student Action**
You get the Hats off for your excellent work!	*Make sure you have given at least 2 Hats off during the class for:* Creatively Solved Activities +10

WhiteHat Jr Live Online Coding for Kids

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Great Question +10

Strong Concentration +10

Project Overview

E-RIDER STAGE 1

Goal of the Project:

In Class 68 you started working on a new case study - a wireless library app. You used bottom navigation to navigate between the search screen and the issue/return screen. In this project, you have to build bottom navigation for an e-rider App, which lets the user navigate between two screens—Rent a Ride & Ride History.

Story:

People are becoming health conscious and looking out for better and affordable options to exercise outdoors. Your friend Vihaan has thought of an idea to start giving bicycles on rent. He has approached you to make an app for the same.

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

Bye Bye!

Students engage with the teacher over the project.

Teacher ends slideshow

WhiteHat Jr Live Online Coding for Kids

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

WhiteHat Jr
Live online Coding for Kids

Teacher Clicks ✖ End Class	
ADDITIONAL ACTIVITIES	
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 as a reflection journal.*

Activity	Activity Name	Links
Student Activity 1	Buzzer App	https://snack.expo.dev/@procodingclass/buzzer-app
Student Activity 2	Material Bottom Tab Navigation Documentation	https://reactnavigation.org/docs/material-bottom-tab-navigator
Student Activity 3	Material Top Tab Navigation Documentation	https://reactnavigation.org/docs/material-top-tab-navigator/
Teacher Activity	Solution Reference Code	https://github.com/React-Native-Frontier/PRO-C68-E-Library
Teacher Reference	Visual aid link	https://curriculum.whitehatjr.com/Vis

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

visual aid link		ual+Project+Asset/PRO_VD/BJFC-PRO-V3-C68-withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/fad5d6a1-abcb-4382-91dc-444101715015.pdf

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.