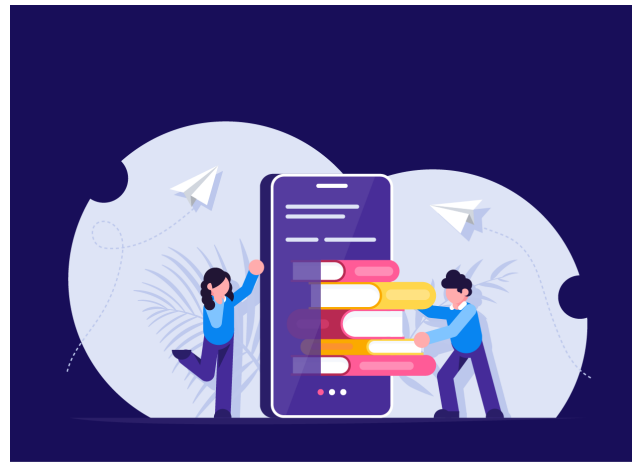


e-library APP



What is our GOAL for this MODULE?

In this class we discussed a skeleton/structure for a library management system app for a low-income school which cannot afford a dedicated librarian.

What did we ACHIEVE in the class TODAY?

- Explored a skeleton/structure of an e-library management system for a low-income school.
- Designed a wireframe for the e-library application.
- Built tab navigation to issue/return and search for a book.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Tab Navigation
- Designing Wireframe

How did we DO the activities?

1. Create a simple library management system app for a low income school.
 - Design a wireframe for the app.
 - Include features like issuing a book, returning a book, checking a book & student eligibility.
2. Use **expo init** command to create a new expo project.
3. Choose a blank template, give a name to the app and wait for all the expo package installations to complete.

```
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.
```

4. Create a screens folder in the app directory called - **screens**.
5. Create two screen components called **SearchScreen** and **BookTransactionScreen** in separate js files in the screens folder.
6. Import all the necessary packages to create the component.

7. Export the component.

```
screens > JS Transaction.js > ...
1  import React, { Component } from "react";
2  import { View, Text, StyleSheet } from "react-native";
3
4  export default class TransactionScreen extends Component {
5    render() {
6      return (
7        <View style={styles.container}>
8          <Text style={styles.text}>Transaction Screen</Text>
9        </View>
10      );
11    }
12  }
13
14  const styles = StyleSheet.create({
15    container: {
16      flex: 1,
17      justifyContent: "center",
18      alignItems: "center",
19      backgroundColor: "#5653D4"
20    },
21    text: {
22      color: "#ffff",
23      fontSize: 30
24    }
25  });
```

```
screens > JS Search.js > ...
1  import React, { Component } from "react";
2  import { View, Text, StyleSheet } from "react-native";
3
4  export default class SearchScreen extends Component {
5    render() {
6      return (
7        <View style={styles.container}>
8          <Text style={styles.text}>Search Screen</Text>
9        </View>
10     );
11   }
12 }
13
14 const styles = StyleSheet.create({
15   container: {
16     flex: 1,
17     justifyContent: "center",
18     alignItems: "center",
19     backgroundColor: "#5653D4"
20   },
21   text: {
22     color: "#ffff",
23     fontSize: 30
24   }
25 });
```

8. Open **App.js** file.
9. Import both the screens and all the necessary react, react-native packages.
10. Import **createBottomTabNavigator** from 'react-navigation-tabs'.
 - **createBottomTabNavigator** will help us create the bottom tab navigator.

```
components > JS BottomTabNavigator.js > ...
1  import React, { Component } from "react";
2  import { NavigationContainer } from "@react-navigation/native";
3  import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
4
5  import TransactionScreen from "../screens/Transaction";
6  import SearchScreen from "../screens/Search";
7
```

11. In the **App.js** file, import the **BottomTabNavigator**
 - Putting it inside a **View** component creates a bug where view overlaps the tab and the tabs are not visible.

```

return (
  <NavigationContainer>
    <Tab.Navigator>
      <Tab.Screen name="Transaction" component={TransactionScreen} />
      <Tab.Screen name="Search" component={SearchScreen} />
    </Tab.Navigator>
  </NavigationContainer>
);

```

- Just like we defined **AppContainer** while creating Switch Navigation, you can create an **AppContainer** containing Tab Navigation.

```

JS App.js > ...
1  import React, { Component } from "react";
2  import BottomTabNavigator from "../components/BottomTabNavigator";
3
4  export default class App extends Component {
5    render() {
6      return <BottomTabNavigator />;
7    }
8  }

```

12. Use **expo start** to run the code.

```

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

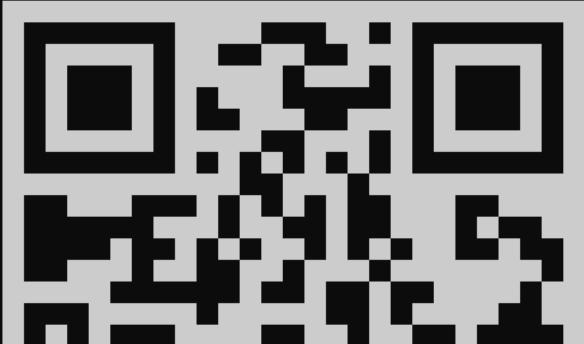
```

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).

```



OUTPUT:



What's NEXT?

In the next classes, we will learn to scan the QR-Code and display the data inside the Text input.

EXTEND YOUR KNOWLEDGE

1. Bookmark this link to know more about Tab Navigation:
<https://reactnavigation.org/docs/tab-based-navigation/>