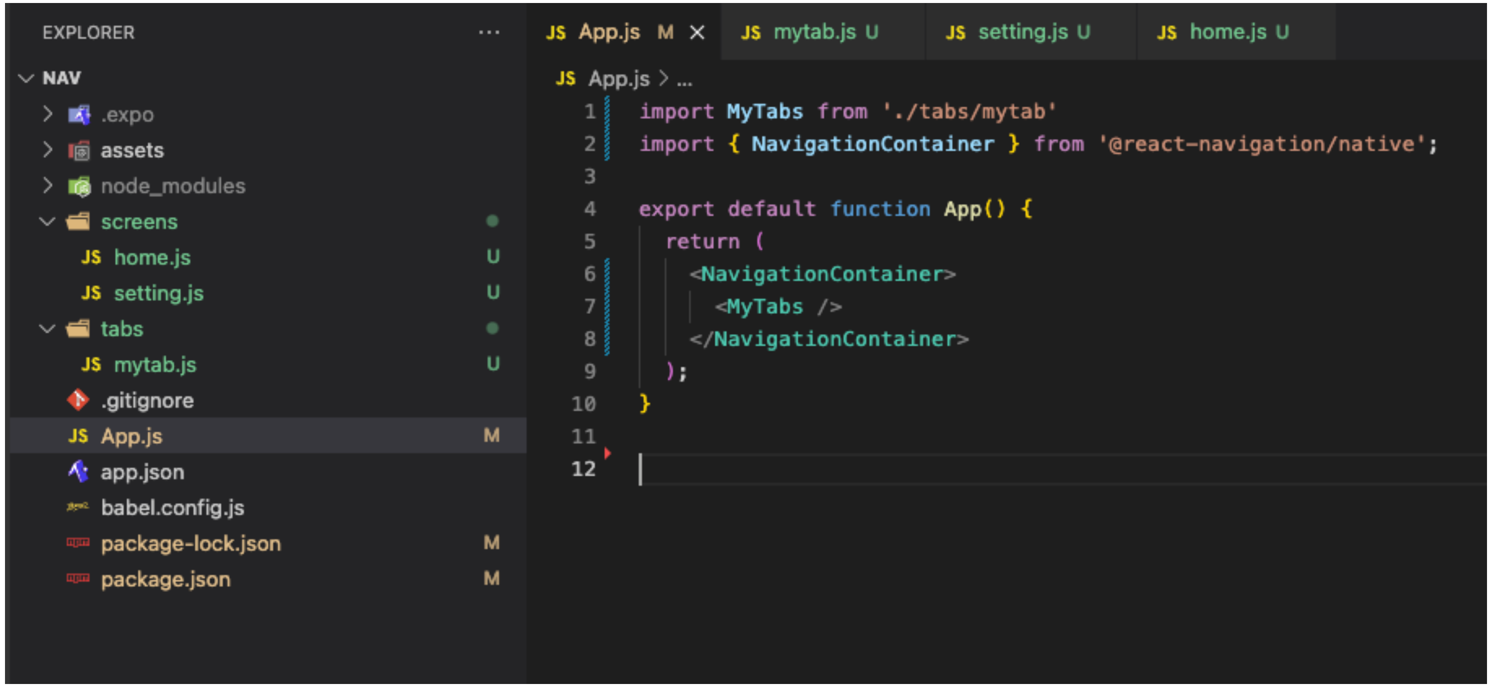
Navigation

Tab navigation using Expo Router

- Possibly the most common style of navigation in mobile apps is tab-based navigation. This can be tabs on the bottom of the screen or on the top below the header (or even instead of a header).
- Creating a new Expo app:
  - *npx create-expo-app@latest –template tabs@50*
  - *npx expo start*

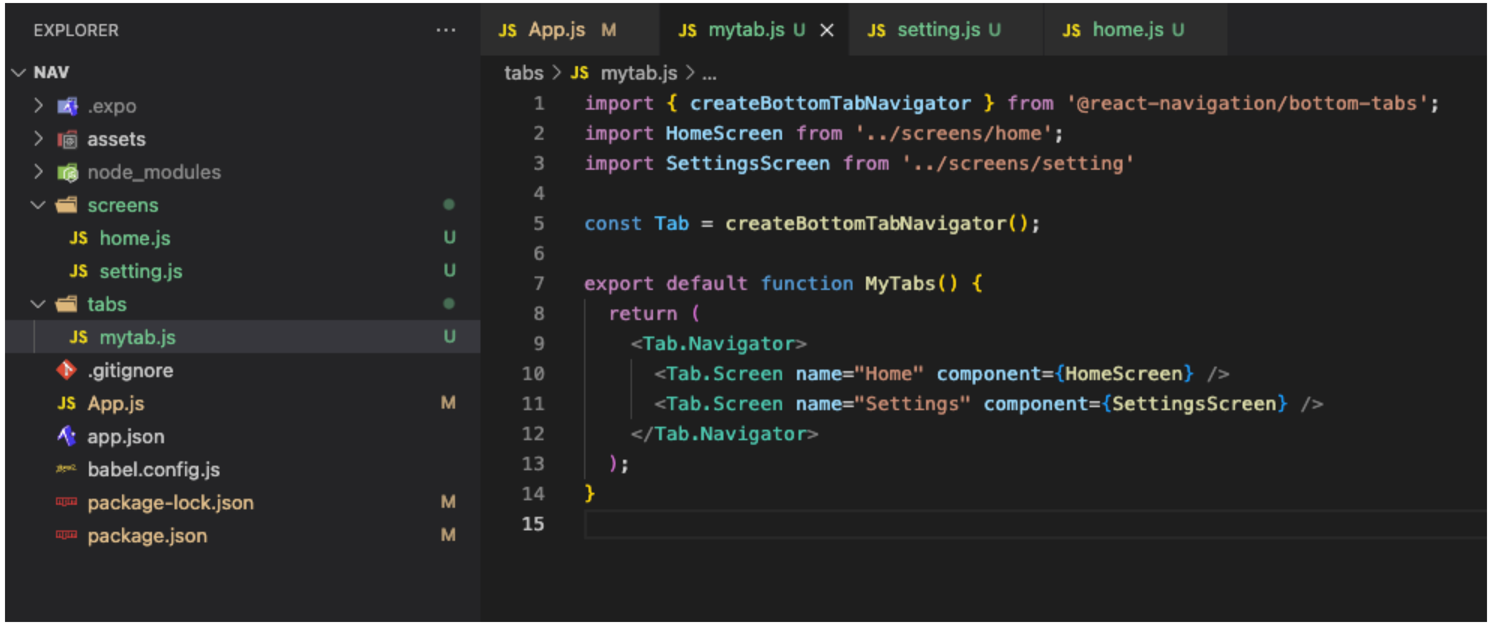
Tab navigation using React Navigation

- Creating a new Expo app:
  - *npx expo install react-native-screens react-native-safe-area-context*
  - *npx expo install @react-navigation/native*
  - *npx expo install @react-navigation/bottom-tabs*
  - *npx expo install @expo/vector-icons*
  - *npx expo start*



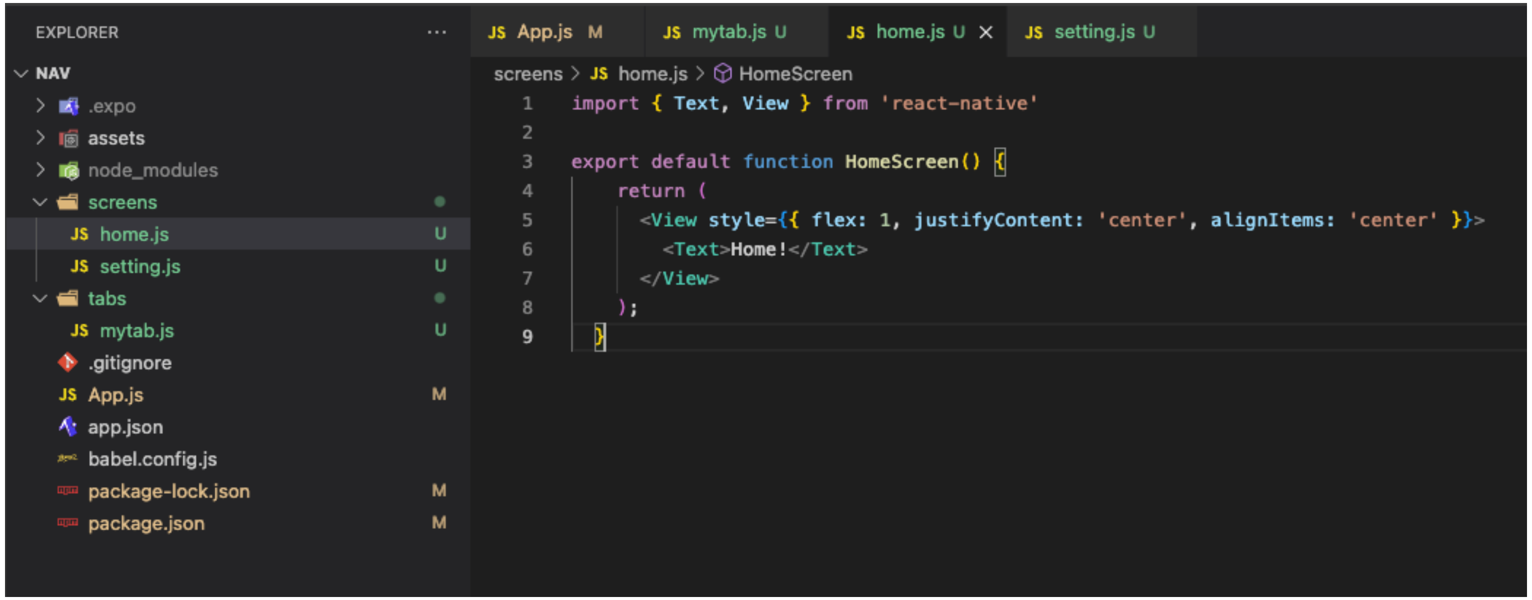
```
import MyTabs from './tabs/mytab'
import { NavigationContainer } from '@react-navigation/native';

export default function App() {
  return (
    <NavigationContainer>
      <MyTabs />
    </NavigationContainer>
  );
}

```

App.js



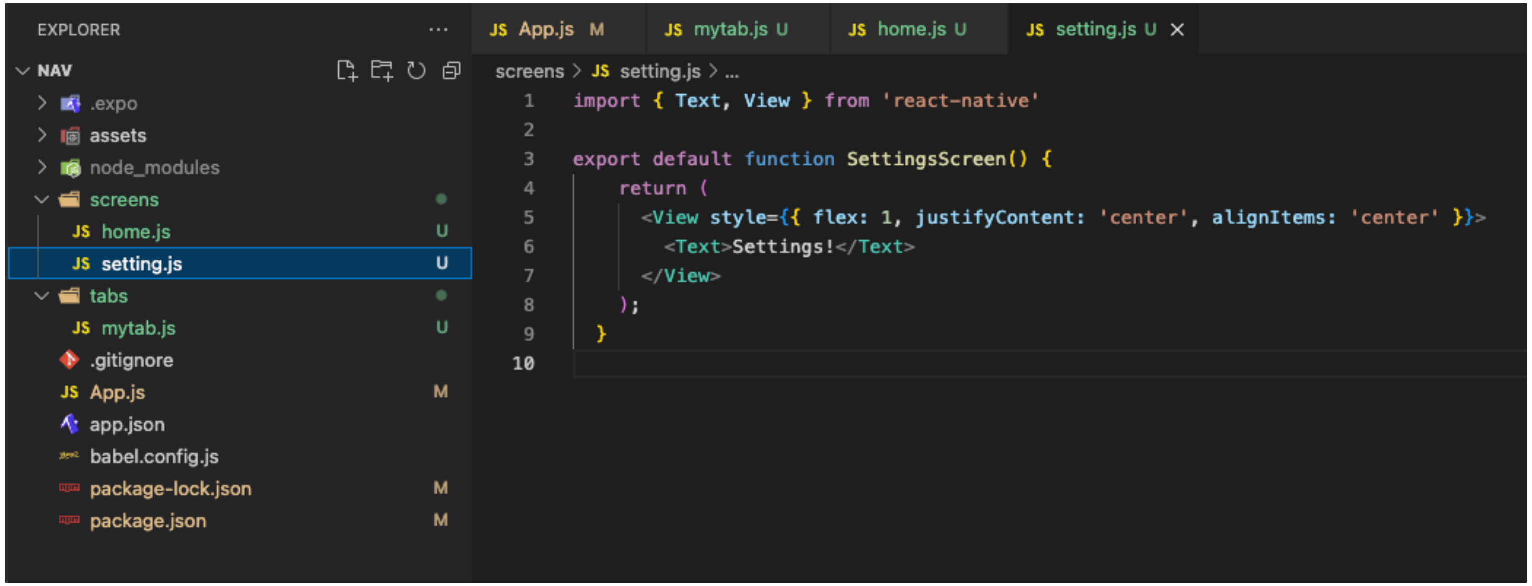
```
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import HomeScreen from '../screens/home';
import SettingsScreen from '../screens/setting'

const Tab = createBottomTabNavigator();

export default function MyTabs() {
  return (
    <Tab.Navigator>
      <Tab.Screen name="Home" component={HomeScreen} />
      <Tab.Screen name="Settings" component={SettingsScreen} />
    </Tab.Navigator>
  );
}

```

mytab.js



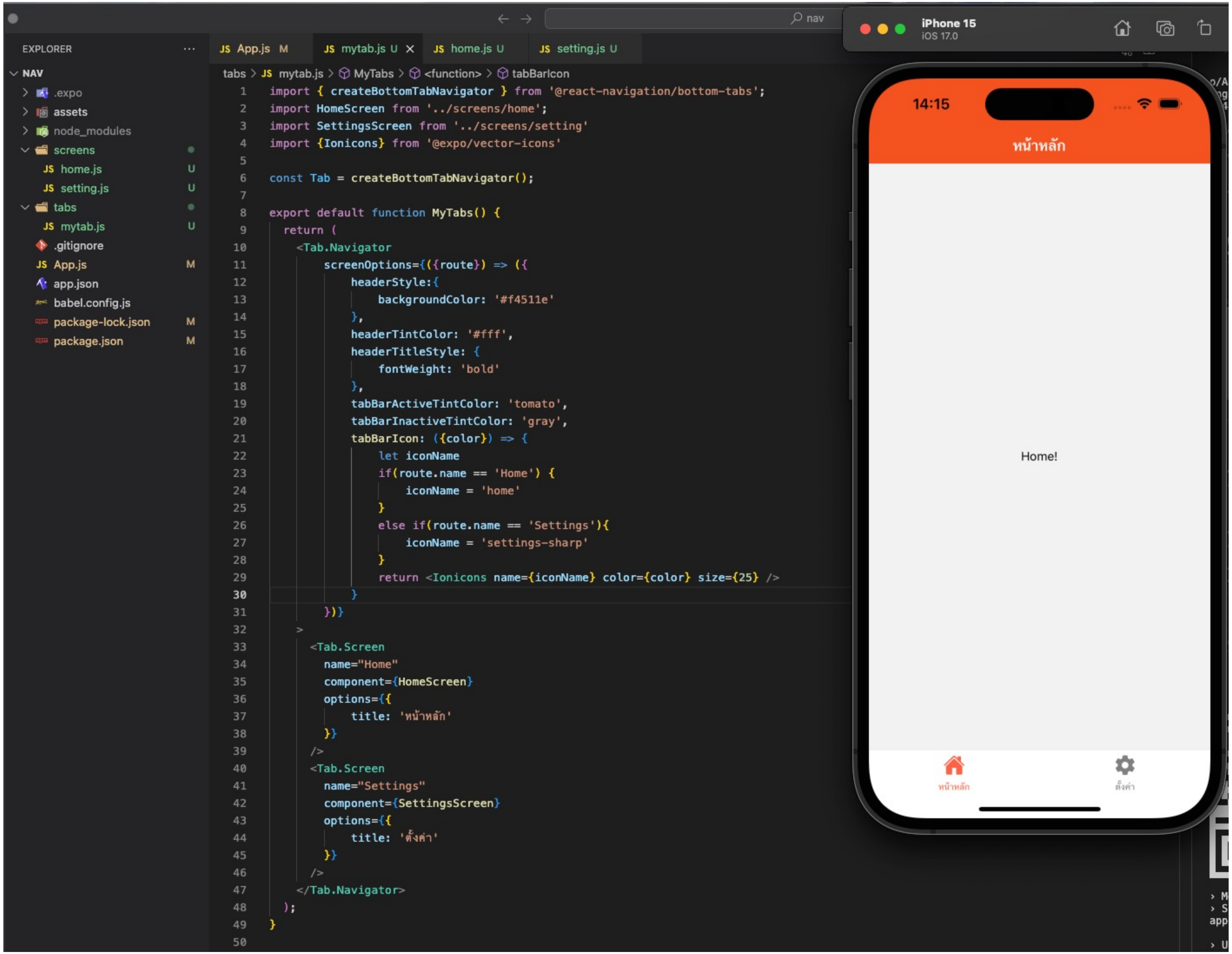
```
import { Text, View } from 'react-native'

export default function HomeScreen() {
  return (
   <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
    <Text>Home!</Text>
   </View>
  );
 }
```

home.js



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

export default function SettingsScreen() {
  return (
   <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
    <Text>Settings!</Text>
   </View>
  );
 }

```

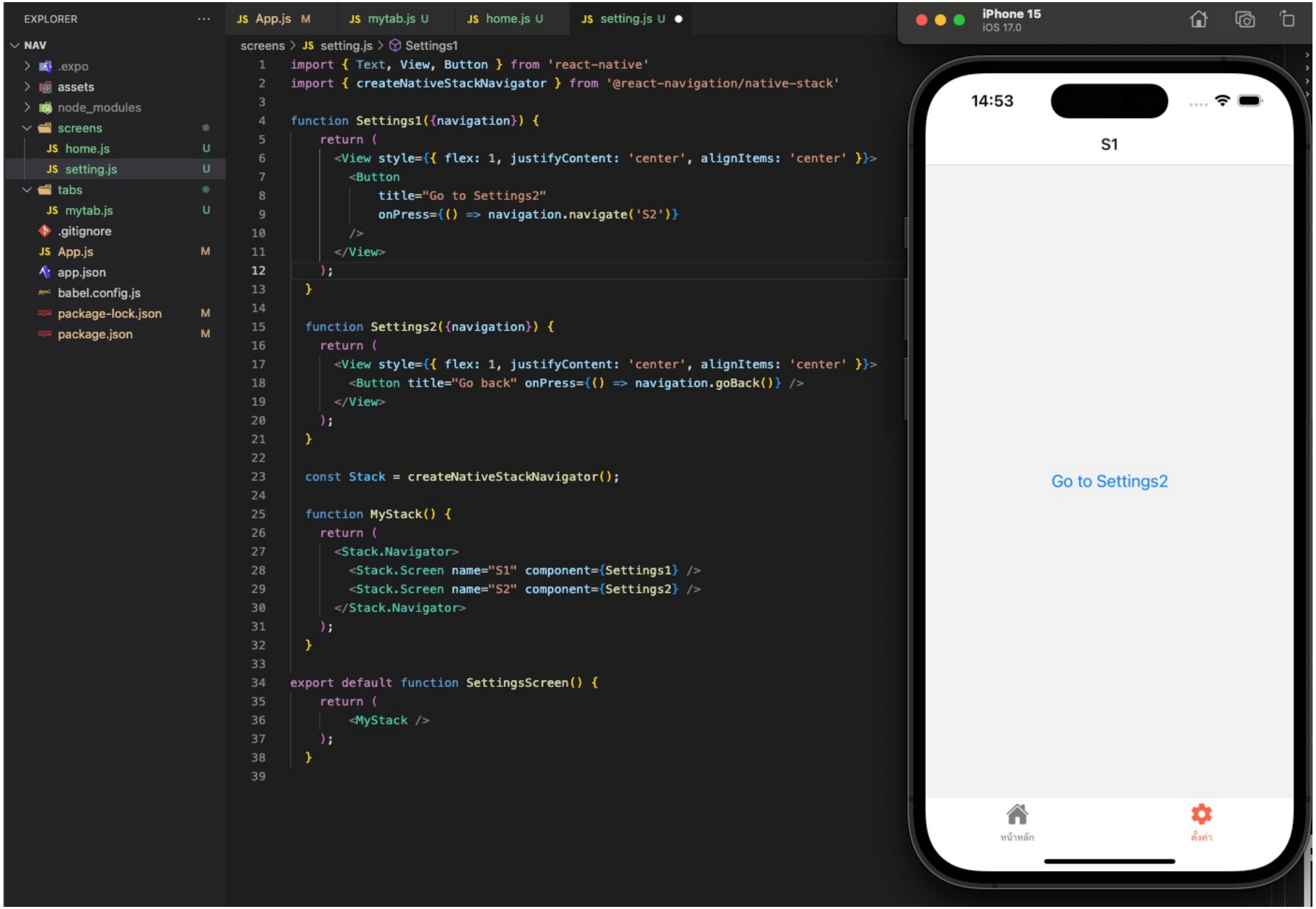
setting.js



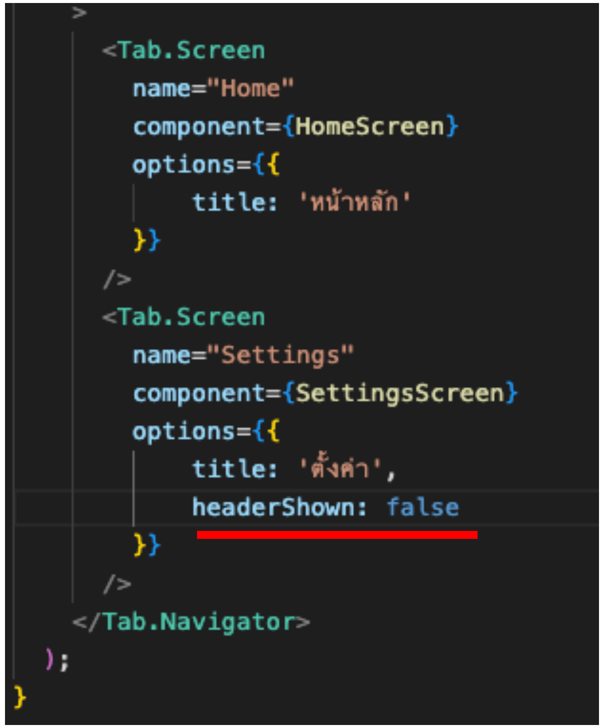
```
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import HomeScreen from '../screens/home';
import SettingsScreen from '../screens/setting'
import {Ionicons} from '@expo/vector-icons'

const Tab = createBottomTabNavigator();

export default function MyTabs() {
  return (
    <Tab.Navigator
        screenOptions={({route}) => ({
            headerStyle:{
                backgroundColor: '#f4511e'
            },
            headerTintColor: '#fff',
            headerTitleStyle: {
                fontWeight: 'bold'
            },
            tabBarActiveTintColor: 'tomato',
            tabBarInactiveTintColor: 'gray',
            tabBarIcon: ({color}) => {
                let iconName
                if(route.name == 'Home') {
                    iconName = 'home'
                }
                else if(route.name == 'Settings'){
                    iconName = 'settings-sharp'
                }
                return <Ionicons name={iconName} color={color} size={25} />
            }
        })}
    >
      <Tab.Screen
        name="Home"
        component={HomeScreen}
        options={{
            title: 'หน้าหลัก'
        }}
      />
      <Tab.Screen
        name="Settings"
        component={SettingsScreen}
        options={{
            title: 'ตั้งค่า'
        }}
      />
    </Tab.Navigator>
  );
}

```

Stack navigation using React Navigation

- Stack Navigator provides a way for your app to transition between screens where each new screen is placed on top of a stack.
- Installation:
  - *npx expo install @react-navigation/native-stack*



```
import { Text, View, Button } from 'react-native'
import { createNativeStackNavigator } from '@react-navigation/native-stack'

function Settings1({navigation}) {
  return (
   <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
    <Button
      title="Go to Settings2"
      onPress={() => navigation.navigate('S2')}
    />
   </View>
  );
 }

 function Settings2({navigation}) {
  return (
   <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
    <Button title="Go back" onPress={() => navigation.goBack()} />
   </View>
  );
 }

 const Stack = createNativeStackNavigator();

 function MyStack() {
  return (
   <Stack.Navigator>
    <Stack.Screen name="S1" component={Settings1} />
    <Stack.Screen name="S2" component={Settings2} />
   </Stack.Navigator>
  );
 }

export default function SettingsScreen() {
  return (
    <MyStack />
  );
 }
```



```
      >
      <Tab.Screen
        name="Home"
        component={HomeScreen}
        options={{
            title: 'หน้าหลัก'
        }}
      />
      <Tab.Screen
        name="Settings"
        component={SettingsScreen}
        options={{
            title: 'ตั้งค่า',
            headerShown: false
        }}
      />
    </Tab.Navigator>
  );
}
```