

LAB 6

MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Chuyển trang trong ứng dụng
- ✓ Tạo Drawer menu và tùy chỉnh giao diện drawer

NỘI DUNG

BÀI 1: TẠO STACK NAVIGATOR CHO ỨNG DỤNG



Màn hình chính



Màn hình chi tiết

Yêu cầu:

- Xây dựng giao diện như trên hình
- Cho người dùng nhập tên ở màn hình chính, sau đó truyền tên nhập thông qua params khi người dùng nhấn nút “Đi tới màn hình chi tiết”
- Ở màn hình chi tiết, sử dụng 4 hàm **goBack**, **reset**, **pop**, **popToTop** của navigator để trở lại màn hình chính.

Hướng dẫn:

- Tạo stack navigator như bao gồm màn hình Home, Details:

```
export type RootStackParamList = {
  Home: undefined;
  Details: {id: number; name: string};
};

const Stack = createNativeStackNavigator<RootStackParamList>();

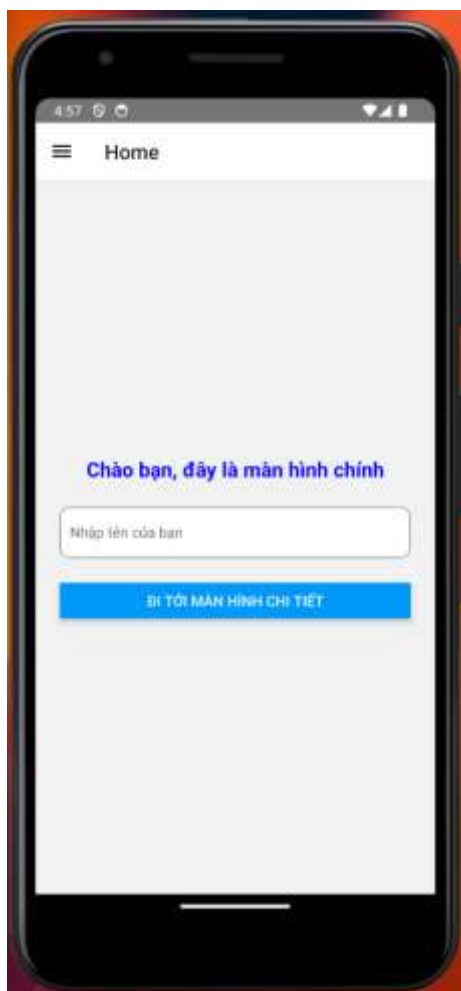
const Bail = () => {
  return (
    <Stack.Navigator screenOptions={{headerShown: false}}>
      <Stack.Screen name="Home" component={HomeScreen} />
      <Stack.Screen name="Details" component={DetailsScreen} />
    </Stack.Navigator>
  );
};
```

- Ở màn hình chính, các bạn tạo một TextInput để nhập tên, sau đó sử dụng tên để truyền qua params của màn hình chi tiết.
- Ở màn hình chi tiết, hiển thị tên và id nhận được từ params màn hình chính. Tạo thêm 4 nút để cho user trở lại màn hình chính, sử dụng các hàm khác nhau của **navigator** để trở lại màn hình chính:

```
const DetailsScreen: FC<
  NativeStackScreenProps<RootStackParamList, 'Details'>
> = ({route, navigation}) => {
  const {id, name} = route.params;
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Chào bạn, {name} </Text>
      <Text style={styles.subText}>Id của bạn là, {id} </Text>

      <Button title="Trở lại bằng -- goBack --" onPress={navigation.goBack} />
      <Button
        title="Trở lại bằng -- reset --"
        onPress={() => navigation.reset({routes: [{name: 'Home'}]})}
      />
      <Button title="Trở lại bằng -- pop --" onPress={() => navigation.pop()} />
      <Button
        title="Trở lại bằng -- popToTop --"
        onPress={() => navigation.popToTop()}
      />
    </View>
  );
};
```

BÀI 2: XÂY DỰNG DRAWER NAVIGATION



Ở màn hình chính



Màn hình chi tiết

Yêu cầu:

- Tạo Drawer Navigator.
- lồng Drawer Navigator vào Stack Navigator các bạn đã làm ở bài trước.

Hướng dẫn:

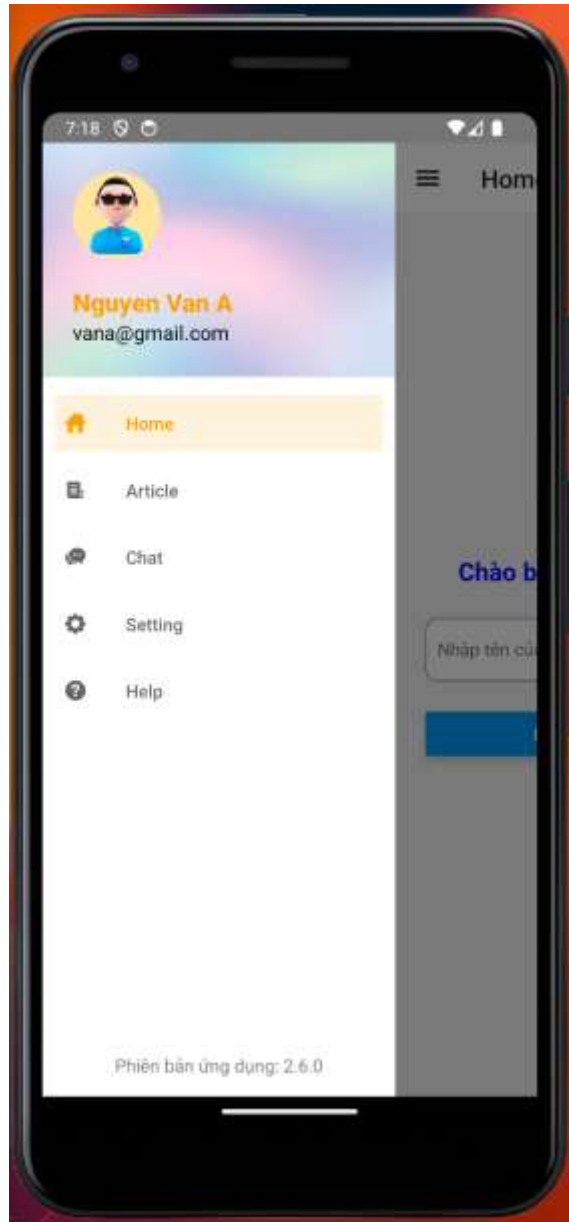
- Xây dựng Drawer Navigator bao gồm 4 menu:

```
const Bai2 = () => {  
  return (  
    <Drawer.Navigator>  
      <Drawer.Screen name="Home" component={HomeScreen} />  
      <Drawer.Screen name="Article" component={ContactScreen} />  
      <Drawer.Screen name="Chat" component={ContactScreen} />  
      <Drawer.Screen name="Setting" component={ContactScreen} />  
    </Drawer.Navigator>  
  );  
};
```

- lồng Drawer Navigator vào Stack Navigator chúng ta đã xây dựng ở bài trước:

```
const Bai1 = () => {  
  return (  
    <Stack.Navigator screenOptions={{headerShown: false}}>  
      <Stack.Screen name="Drawer" component={Bai2} />  
      <Stack.Screen name="Home" component={HomeScreen} />  
      <Stack.Screen name="Details" component={DetailsScreen} />  
    </Stack.Navigator>  
  );  
};
```

BÀI 3: TÙY CHỈNH GIAO DIỆN CHO DRAWER NAVIGATOR



Hướng dẫn:

- Để chèn thêm icon sử dụng **drawerIcon** trong **options** của **Drawer.Screen**.
- Để tùy chỉnh header và footer cho drawer các bạn có thể sử dụng **drawerContent** trong **Drawer.Navigator**.

BÀI 4: GV CHO THÊM

***** YÊU CẦU NỘP BÀI:**

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết