

**VIETNAM NATIONAL UNIVERSITY - HO CHI MINH
UNIVERSITY OF SCIENCE**

Coursera - React Native of Meta

Module 4

Academic Supervisors:

Student:

23127438 - Dang Truong Nguyen

Bui Duy Dang

Huynh Lam Hai Dang

Tran Trung Kien

Semester: I

Academic Year: 2025 - 2026

Ho Chi Minh City, November 4, 2025

TABLE OF CONTENTS

TABLE OF CONTENTS	1
LIST OF FIGURES	2
1 React Navigation	3
1.1 Video: What is React Navigation	3
1.1.1 Problem	3
1.1.2 React Native Navigation	3
1.1.3 Benefit of using React Native Navigation	3
1.1.4 Trade off	4
1.1.5 Other commonly used library	4
1.2 Reading: Installation and Setup of React Navigation	4
1.3 Video: Setting up React Navigation	5
1.4 Video: Using the Stack.Navigator	5
1.5 Reading: Creating and Configuring Native Stack Navigator	5
1.6 Video: Approaches to Passing Props to Screen	6
1.7 Reading: Passing props to Screen	6
1.8 Reading: Exercise: Set up Routes	7
1.9 Practice Assignment: Self review: Set up Routes	7
1.10 Practice Assignment: Knowledge Check: Introduction to React Navigation	8

LIST OF FIGURES

Figure 1.1	Using the Stack.Navigator	5
Figure 1.2	Passing prop into Stack	6
Figure 1.3	Exercise: Set up Routes	7
Figure 1.4	Self review: Set up Routes	7
Figure 1.5	Knowledge Check: Introduction to React Navigation	8

CHAPTER 1

React Navigation

1.1. Video: What is React Navigation

1.1.1. Problem

- Moving between screens is referred to as navigation or routing in mobile apps.
- Mobile apps have multiple screens and the user's experience relies on smooth navigation between these screens.
- React Native does not have inbuilt navigation, it must pick a navigation library to implement moving between the screens and the app. → React Navigation comes onto the screen

1.1.2. React Native Navigation

- React Navigation is the most popular navigation library for React Native apps today.
- It is easy to use, customizable based on your needs, and has platform-specific components for both iOS and Android.
- Can be used for both React Native CLI apps and React Native built with Expo.
- It provides a Native Stack Navigator that can transition between screens and manage the navigation history.

1.1.3. Benefit of using React Native Navigation

- The difference between routing and navigation works in web browsers versus React Navigation for mobile app: **React Navigation provides the gestures and animations.**
- React Native provides Screens, which provide native navigation container components

- Its also provides gesture handler, which make React Native more compatible with native touch and gesture system.
- Which brings:
 - Easier to push OTA updates.
 - Easier to debug.
 - Possible to customize navigator components.

1.1.4. Trade off

- Some of the navigators don't directly use the native navigation APIs on iOS and Android.
- While gaining access to new features after upgrading is generally good news, it's possible that code using older APIs will no longer work as expected.

1.1.5. Other commonly used library

- React-native-router-flux
- React-native-navigation

Practice Question: React Navigation is the most popular navigation library for React Native apps. Which of the following are characteristics of this library? Select all that apply. → Answer:

- iOS-specific components.
- Customizable navigators
- Android-specific components.

1.2. Reading: Installation and Setup of React Navigation

```
npm install @react-navigation/native  
  
npx expo install react-native-screens  
      react-native-safe-area-context
```

1.3. Video: Setting up React Navigation

Documentation: <https://reactnavigation.org/> **Practice Question:** What are the base packages you would need to install to setup React Navigation within your project? Select all that apply. → Answer:

- @react-navigation/native
- react-native-safe-area-context
- react-native-screens

1.4. Video: Using the Stack.Navigator

Practice Question: The name given to a screen within a Stack.Navigator must match the name of the component that it calls. True or false? → Answer: False

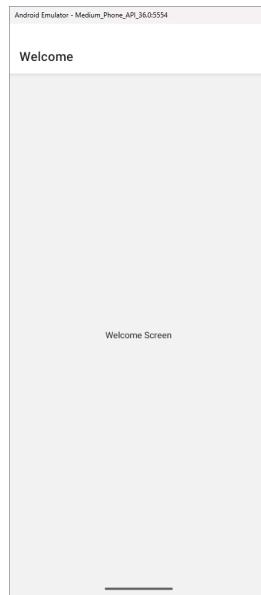


Figure 1.1: Using the Stack.Navigator

1.5. Reading: Creating and Configuring Native Stack Navigator

This part is the same progress of using Stack.Navigator that I have done in the above section.

1.6. Video: Approaches to Passing Props to Screen

- To change the title of the Stack Screen, use in Stack.Screen:

```
options={{ title: '...' }}
```

- To set the root Stack that would be viewed when user open the first time, use in Stack.Navigator:

```
initialRouteName='...'
```

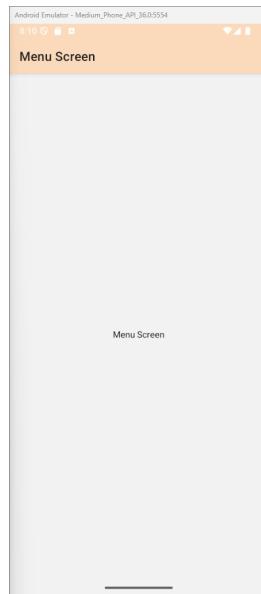


Figure 1.2: Passing prop into Stack

1.7. Reading: Passing props to Screen

This part is the same progress that I have done in the above section.

1.8. Reading: Exercise: Set up Routes

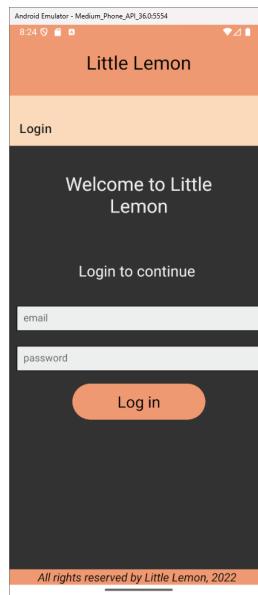


Figure 1.3: Exercise: Set up Routes

1.9. Practice Assignment: Self review: Set up Routes

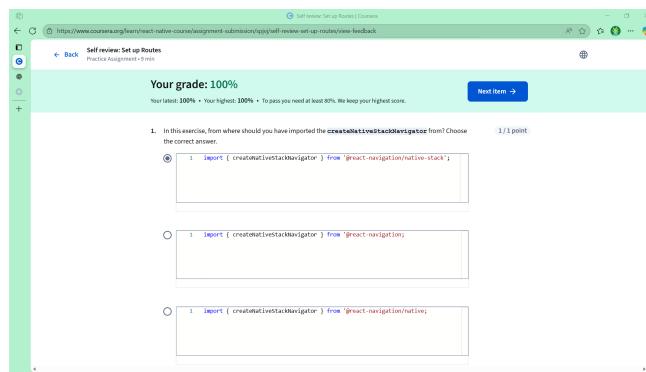


Figure 1.4: Self review: Set up Routes

1.10. Practice Assignment: Knowledge Check: Introduction to React Navigation

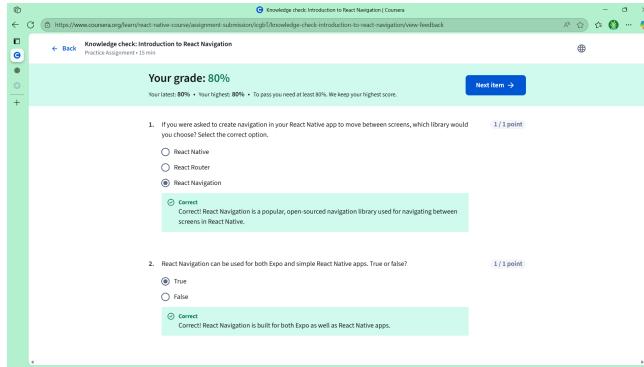


Figure 1.5: Knowledge Check: Introduction to React Navigation