```
sudo npm install -g react-native-cli
```

After that, we need to generate the project and go into that directory.

```
react-native init RNNavigation
cd RNNavigation
```

Now, I am showing you this example on the iOS device so; we need to start the ios development. Hit the following command to generate iOS build.

```
react-native run-ios
```

# Create two screens for our project.

In the root directory, create one folder called **screens.** Go into that folder and create two screens.

1. **Settings.js**
2. **Home.js**

Now, add the following code to **Settings.js** file.

```
// Settings.js

import React, { Component } from 'react';
import { View, Text, Button } from 'react-native';

export class Settings extends Component {
  render() {
    return (
      <View>
        <Text>This is the Settings screen</Text>
      </View>
    )
  }
};

export default Settings;
```

After that add the following code to **Home.js** file.

```
// Home.js

import React, { Component } from 'react';
import { View, Text } from 'react-native';

export class Home extends Component {
  render() {
    return (
      <View>
        <Text>This is the home screen</Text>
      </View>
    )
  }
}

export default Home
```

So, now we have both screen ready. We just need to add navigation to it.

# Install React Navigation package.

Go to your project root and in the terminal, add the package using the following command.

```
yarn add react-navigation
```

Now, add react navigation module inside **App.js** file.

```
// App.js

import { StackNavigator } from 'react-navigation';
```

**StackNavigator** provides the way for your app to transition between the screens and manage navigation history.

Now, import the two screens we have created inside the **App.js** file.

```
// App.js

import Settings from './screens/Settings';
import Home from './screens/Home';
```

Now, pass both the screen on the **StockNavigator** function.

```
// App.js

const AppNavigator = StackNavigator({
  SettingScreen: { screen: Settings },
  HomeScreen: { screen: Home }
});
```

So, our final **App.js** file looks like this.

```
// App.js

import React, { Component } from 'react';
import {
  Platform,
  StyleSheet,
  Text,
  View
} from 'react-native';
import { StackNavigator } from 'react-navigation';
import Settings from './screens/Settings';
import Home from './screens/Home';

const AppNavigator = StackNavigator({
  SettingScreen: { screen: Settings },
  HomeScreen: { screen: Home }
});

export default class App extends Component {
  render() {
```

```
    return (
      <AppNavigator />
    );
  }
}
```

## Add the Navigation Button inside Settings.js file.

```
// Settings.js

export class Settings extends Component {
  render() {
    return (
      <View>
        <Text>This is the Settings screen</Text>
        <Button onPress={() => this.props.navigation.navigate('HomeScreen')} title="H
ome"/>
      </View>
    )
  }
};
```

So, what we have done is when the user clicks that button, it will redirect to **HomeScreen,** which we have defined inside the **App.js** file. So our screen smoothly changes to the clicked one. In our case it is **HomeScreen**. Our final **Settings.js** file looks like this.

```
// Settings.js

import React, { Component } from 'react';
import { View, Text, Button } from 'react-native';

export class Settings extends Component {
  render() {
    return (
```

```
      <View>
        <Text>This is the Settings screen</Text>
        <Button onPress={() => this.props.navigation.navigate('HomeScreen')} title="H
ome"/>
      </View>
    )
  }
};

export default Settings;
```

# Reload the application.

Now, reload the iPhone Simulator. You will see the screen like this.