Cross Platform App Development Lab Experiment No.7

Aim: Implementing multiple screen navigation and nested navigation using solutions provided by React Navigation and .NET MAUI.

Objectives:

1. Implement multiple screen navigation in a React Native application.
2. Explore the solutions provided by React Navigation for nested navigation.
3. Implement multiple screen navigation in a .NET MAUI application.
4. Utilize .NET MAUI's solutions for nested navigation.

Theory:

- React Navigation:

- Library for navigation in React Native applications.
- Supports stack, tab, drawer, and other types of navigators.
- Allows easy management of multiple screens.
- Nested Navigation in React Navigation:
- Involves navigating within screens of a navigator.
- Achieved using nested navigators like Stack Navigator within Tab Navigator.

- .NET MAUI Navigation:

- .NET Multi-platform App UI (MAUI) framework for cross-platform app development.
- Supports navigation patterns similar to Xamarin Forms.
- Offers navigation containers and pages for screen navigation.
- Nested Navigation in .NET MAUI:
- Involves navigating between pages and utilizing navigation containers.
- Hierarchical navigation structure for managing nested navigation.

Requirements:

- React Native development environment for React Navigation.
- .NET MAUI development environment for .NET MAUI navigation.

Tools:

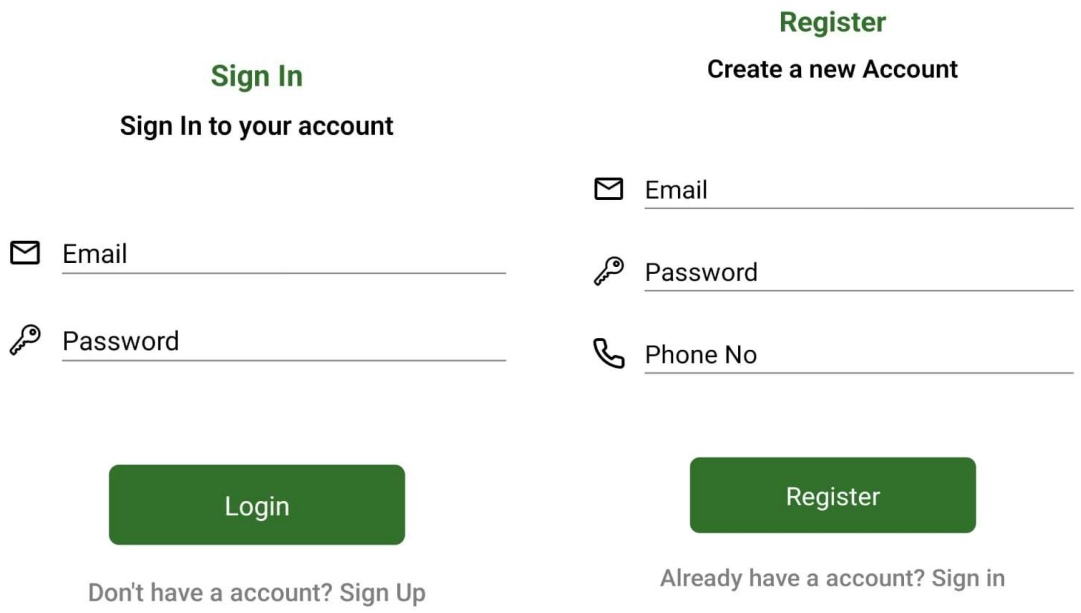
- React Navigation library for React Native.

- Visual Studio or Visual Studio Code for .NET MAUI development.

Implementation/ Code:-

```
import { StyleSheet, Text, View } from 'react-native'
import React from 'react'
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import { NavigationContainer } from '@react-navigation/native';
import HomeScreen from './screens/HomeScreen';
import PickUpScreen from './screens/PickUpScreen';
import CartScreen from './screens/CartScreen';
import LoginScreen from './screens/LoginScreen';
import RegisterScreen from './screens/RegisterScreen';
import ProfileScreen from './screens/ProfileScreen';


const StackNavigator = () => {
  const Stack = createNativeStackNavigator();
return (
  <NavigationContainer>
    <Stack.Navigator>
      <Stack.Screen name="Login" component={LoginScreen}
options={{headerShown:false}}/>
      <Stack.Screen name="Home" component={HomeScreen}
options={{headerShown:false}}/>
      <Stack.Screen name="PickUp" component={PickUpScreen}
options={{headerShown:false}}/>
      <Stack.Screen name="Cart" component={CartScreen}
options={{headerShown:false}}/>

      <Stack.Screen name="Register" component={RegisterScreen}
options={{headerShown:false}}/>
      <Stack.Screen name="Profile" component={ProfileScreen}
options={{headerShown:false}}/>


    </Stack.Navigator>
  </NavigationContainer>
)
}

export default StackNavigator

const styles = StyleSheet.create({})
```

- Used stack navigator for navigating from one screen to another.



- First we, clicked on **Don't have a account? Sign Up** then we are navigated to another page that is **register page**

Conclusion:

By implementing multiple screen navigation using React Navigation for React Native built-in navigation solutions, we enhance the user experience and provide a structured flow within the applications.

References:

1. React Navigation documentation:https://reactnavigation.org/docs/getting-started

2. .NET MAUI documentation: https://docs.microsoft.com/en-us/dotnet/maui