Assignment 3 – Navigation in Apps

CLO-3: Demonstrate the concepts of React Native & React Navigation in the context of mobile application development

Name: Sadaat Ali Taj

Roll No: SP22-BSE-015

Submitted to: Sir Kamran

Submission Date: 23 May, 2025

# Objective

The objective of this assignment is to develop a mobile application using React Native that demonstrates the use of bottom tab navigation. The app must contain three tabs: Home, Profile, and Settings, and must utilize location services and state management for dynamic updates.

# Tools & Technologies Used

- React Native (Expo CLI)  
- React Navigation (Bottom Tabs)  
- Expo Location API  
- JavaScript (ES6+)  
- Android Emulator / Expo Go

# App Description

The application consists of three screens navigable via bottom tab navigation:

## 1. Home Screen

- This screen fetches and displays the current city of the user based on their device’s location.

- It uses Expo Location API to access the GPS and reverse-geocode the coordinates to a readable city name.

## 2. Profile Screen

- Displays a TextInput field where the user can edit their name.

- The screen title dynamically updates to reflect the user’s current name input.

## 3. Settings Screen

- A static screen displaying placeholder content, reserved for future settings implementation.

# Code Implementation

App.js

import React, { useState, useEffect } from 'react';  
import { Text, View, TextInput, StyleSheet } from 'react-native';  
import \* as Location from 'expo-location';  
import { NavigationContainer } from '@react-navigation/native';  
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';  
  
const Tab = createBottomTabNavigator();  
  
function HomeScreen() {  
 const [city, setCity] = useState('Loading...');  
 useEffect(() => {  
 (async () => {  
 let { status } = await Location.requestForegroundPermissionsAsync();  
 if (status !== 'granted') {  
 setCity('Permission denied');  
 return;  
 }  
 let location = await Location.getCurrentPositionAsync({});  
 let geocode = await Location.reverseGeocodeAsync(location.coords);  
 if (geocode.length > 0) {  
 setCity(geocode[0].city);  
 } else {  
 setCity('City not found');  
 }  
 })();  
 }, []);  
 return (  
 <View style={styles.container}>  
 <Text style={styles.text}>Your City: {city}</Text>  
 </View>  
 );  
}  
  
function ProfileScreen({ navigation }) {  
 const [name, setName] = useState('John Doe');  
 useEffect(() => {  
 navigation.setOptions({ title: name });  
 }, [name]);  
 return (  
 <View style={styles.container}>  
 <Text style={styles.text}>Edit your name:</Text>  
 <TextInput  
 style={styles.input}  
 value={name}  
 onChangeText={(text) => setName(text)}  
 />  
 </View>  
 );  
}  
  
function SettingsScreen() {  
 return (  
 <View style={styles.container}>  
 <Text style={styles.text}>Settings will go here</Text>  
 </View>  
 );  
}  
  
export default function App() {  
 return (  
 <NavigationContainer>  
 <Tab.Navigator>  
 <Tab.Screen name="Home" component={HomeScreen} />  
 <Tab.Screen name="Profile" component={ProfileScreen} />  
 <Tab.Screen name="Settings" component={SettingsScreen} />  
 </Tab.Navigator>  
 </NavigationContainer>  
 );  
}  
  
const styles = StyleSheet.create({  
 container: { flex: 1, justifyContent: 'center', alignItems: 'center' },  
 text: { fontSize: 20 },  
 input: {  
 borderWidth: 1,  
 padding: 10,  
 marginTop: 10,  
 width: 200,  
 borderRadius: 5,  
 },  
});