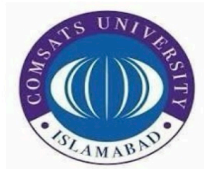


Comsats University Islamabad Attock campus



Name: M.HAROON

Reg no : SP22-BSE-025

Assignment: MAD-04

Submit-To:

MR.MUHAMMAD KAMRAN

CODE#

```
import React, { useState } from 'react';
import { NavigationContainer } from '@react-
navigation/native';
import { createStackNavigator } from '@react-
navigation/stack';
import { TextInput, Button, Text, View, StyleSheet } from
'react-native';

// Signup Screen
const SignupScreen = ({ navigation }) => {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [email, setEmail] = useState("");
  const [phone, setPhone] = useState("");

  const validateForm = () => {
    if (username.match(/^[a-zA-Z]/)) {
      alert('Username should contain alphabets only');
      return false;
    }
    if (password.length < 6) {
      alert('Password should be at least 6 characters long');
      return false;
    }
    if (!email.includes('@')) {
      alert('Please enter a valid email');
      return false;
    }
    if (!phone.match(/^\+92-3\d{2}-\d{7}$/)) {
      alert('Phone number should match +92-3xx-xxxxxxx');
      return false;
    }
  }
}
```

```

    }
    return true;
  };

  const handleSignup = () => {
    if (validateForm()) {
      navigation.navigate('Login');
    }
  };

  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Username"
        value={username}
        onChangeText={setUsername}
      />
      <TextInput
        style={styles.input}
        placeholder="Password"
        secureTextEntry
        value={password}
        onChangeText={setPassword}
      />
      <TextInput
        style={styles.input}
        placeholder="Email"
        value={email}
        onChangeText={setEmail}
      />
      <TextInput
        style={styles.input}
        placeholder="Phone Number (+92-3xx-xxxxxxx)"
        value={phone}
        onChangeText={setPhone}
      />
      <Button title="Signup" onPress={handleSignup} />
    </View>
  );
};

// Login Screen

```

```

const LoginScreen = ({ navigation }) => {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  const handleLogin = () => {
    if (username && password) {
      navigation.navigate('Profile');
    } else {
      alert('Please enter both username and password');
    }
  };

  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Username"
        value={username}
        onChangeText={setUsername}
      />
      <TextInput
        style={styles.input}
        placeholder="Password"
        secureTextEntry
        value={password}
        onChangeText={setPassword}
      />
      <Button title="Login" onPress={handleLogin} />
    </View>
  );
};

// Profile Screen
const ProfileScreen = () => {
  return (
    <View style={styles.container}>
      <Text>Welcome to your Profile!</Text>
    </View>
  );
};

// Navigation setup
const Stack = createStackNavigator();

```

```

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Signup">
        <Stack.Screen name="Signup"
          component={SignupScreen} />
        <Stack.Screen name="Login" component={LoginScreen}
        />
        <Stack.Screen name="Profile"
          component={ProfileScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

// Styles for the screens
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    padding: 20,
  },
  input: {
    height: 40,
    borderColor: 'gray',
    borderWidth: 1,
    marginBottom: 10,
    paddingLeft: 10,
  },
});

export default App;

```

EXPLANATION

1. Imports and Dependencies

- **React and useState:** The core library for building user interfaces in React Native and the useState hook for managing state within functional components.
- **@react-navigation/native and @react-navigation/stack:** These libraries are used to set up navigation between screens in the application.
- **React Native components like TextInput, Button, Text, View, and StyleSheet:** These are UI components provided by React Native for building the app interface.

2. Signup Screen

- **Purpose:** This screen allows users to sign up by providing a username, password, email, and phone number.
- **State Variables:**
 - username, password, email, and phone store the user's input.
- **Form Validation:**
 - Checks if the username contains only alphabets.
 - Ensures the password is at least 6 characters long.
 - Validates the email format by checking the presence of @.
 - Matches the phone number against the pattern +92-3xx-xxxxxxx.
- **Handle Signup:**
 - Calls validateForm() and navigates to the Login screen if validation passes.

3. Login Screen

- **Purpose:** Allows users to log in with their username and password.
- **State Variables:**
 - username and password store the user's input.
- **Handle Login:**
 - Validates that both fields are not empty.
 - Navigates to the Profile screen if validation passes.
 - Alerts the user if fields are missing.

4. Profile Screen

- **Purpose:** A simple screen that welcomes the user after successful signup and login.
- **UI:** Contains only a text message saying, "Welcome to your Profile!".

5. Navigation Setup

- **Stack Navigator:**
 - A Stack.Navigator is used to define the navigation flow between the Signup, Login, and Profile screens.
 - The initial route is set to Signup.
- **Screens:**
 - SignupScreen, LoginScreen, and ProfileScreen are registered as stack screens.

6. App Component

- **Purpose:** The root component of the application.
- **Structure:** Wraps the Stack.Navigator inside a NavigationContainer, which manages the navigation state of the app.

7. Styles

- **Container:**
 - Applies flex layout, centers content vertically, and adds padding to the screens.
- **Input:**
 - Styles for the TextInput elements, including height, border styling, and padding for text input.

8. Key Features

- **Navigation:** Smooth navigation between screens using the react-navigation library.
- **Form Validation:** Ensures user inputs meet the required criteria.
- **State Management:** Uses useState for local state handling.
- **User Experience:** Alerts for invalid inputs and successful navigation feedback.

This code provides a foundational structure for a React Native application with navigation and basic form handling.