

**NAME RIMSHA AQEEL**

**ROLL NO SP22-BSE-040**

```
import React from 'react';
import {
  View,
  Text,
  TextInput,
  TouchableOpacity,
  StyleSheet,
  Alert,
  ScrollView,
} from 'react-native';
import { Provider, useDispatch, useSelector } from 'react-redux';
import { createStore } from 'redux';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import AsyncStorage from '@react-native-async-storage/async-storage';
import * as Location from 'expo-location';

// Firebase API Endpoint
const FIREBASE_API_ENDPOINT =
  'https://reactnativeapp-e9632-default-rtdb.firebaseio.com';

// Redux initial state
const initialState = {
  email: '',
  username: '',
  password: '',
  phone: '',
  location: '',
};

// Redux actions
const UPDATE_SIGNUP = 'UPDATE_SIGNUP';
const SET_LOCATION = 'SET_LOCATION';

const updateSignup = (payload) => ({
  type: UPDATE_SIGNUP,
  payload,
});

const setLocation = (location) => ({
  type: SET_LOCATION,
  location,
});
```

```

// Redux reducer
const reducer = (state = initialState, action) => {
  switch (action.type) {
    case UPDATE_SIGNUP:
      return { ...state, ...action.payload };
    case SET_LOCATION:
      return { ...state, location: action.location };
    default:
      return state;
  }
};

// Redux store
const store = createStore(reducer);

// Stack Navigator
const Stack = createStackNavigator();

// Signup Screen
const SignupScreen = ({ navigation }) => {
  const dispatch = useDispatch();
  const { email, username, password, phone } = useSelector((state) => state);

  const validateSignup = async () => {
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    const usernameRegex = /^[a-zA-Z]+$/;
    const phoneRegex = /^\+92-3\d{2}-\d{7}$/;

    if (!emailRegex.test(email)) {
      Alert.alert('Invalid Email', 'Please enter a valid email address.');
```

```

    );
    return;
}

// Store user data in Firebase
const userData = { email, username, password, phone };
try {
    await fetch(`${FIREBASE_API_ENDPOINT}/users.json`, {
        method: 'POST',
        body: JSON.stringify(userData),
        headers: {
            'Content-Type': 'application/json',
        },
    });
    Alert.alert('Success', 'User registered successfully!');
    navigation.navigate('Login');
} catch (error) {
    Alert.alert('Error', 'Could not store user data.');
```

```

};

return (
    <View style={styles.container}>
        <Text style={styles.title}>Signup</Text>
        <TextInput
            placeholder="Email"
            style={styles.input}
            value={email}
            onChangeText={(value) => dispatch(updateSignup({ email: value })))}
        />
        <TextInput
            placeholder="Username"
            style={styles.input}
            value={username}
            onChangeText={(value) => dispatch(updateSignup({ username: value })))}
        />
        <TextInput
            placeholder="Password"
            secureTextEntry
            style={styles.input}
            value={password}
            onChangeText={(value) => dispatch(updateSignup({ password: value })))}
        />
        <TextInput
            placeholder="Phone (+92-3xx-xxxxxxx)"
```

```

        style={styles.input}
        value={phone}
        onChangeText={(value) => dispatch(updateSignup({ phone: value })))}
      />
      <TouchableOpacity style={styles.button} onPress={validateSignup}>
        <Text style={styles.buttonText}>Signup</Text>
      </TouchableOpacity>
    </View>
  );
};

// Login Screen
const LoginScreen = ({ navigation }) => {
  const { email, password } = useSelector((state) => state);
  const dispatch = useDispatch();

  const validateLogin = async () => {
    if (email === '' || password === '') {
      Alert.alert('Error', 'Please fill in all fields.');
```

```

        placeholder="Email"
        style={styles.input}
        value={email}
        onChangeText={(value) => dispatch(updateSignup({ email: value })))}
      />
      <TextInput
        placeholder="Password"
        secureTextEntry
        style={styles.input}
        value={password}
        onChangeText={(value) => dispatch(updateSignup({ password: value })))}
      />
      <TouchableOpacity style={styles.button} onPress={validateLogin}>
        <Text style={styles.buttonText}>Login</Text>
      </TouchableOpacity>
    </View>
  );
};

```

*// Profile Screen*

```

const ProfileScreen = () => {
  const { email, username, phone, location } = useSelector((state) => state);

  return (
    <ScrollView style={styles.container}>
      <Text style={styles.title}>My Profile</Text>
      <View style={styles.profileCard}>
        <Text>Email: {email}</Text>
        <Text>Username: {username}</Text>
        <Text>Phone: {phone}</Text>
        <Text>Location: {location}</Text>
      </View>
    </ScrollView>
  );
};

```

*// Main App Component*

```

const App = () => {
  return (
    <Provider store={store}>
      <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>
    </Provider>
  );
};

```

```
        </Stack.Navigator>
      </NavigationContainer>
    </Provider>
  );
};

// Styles
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f5f5f5',
    paddingHorizontal: 20,
    paddingVertical: 10,
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
  },
  input: {
    borderWidth: 1,
    borderColor: '#ddd',
    padding: 10,
    marginBottom: 15,
    borderRadius: 5,
    backgroundColor: '#fff',
  },
  button: {
    backgroundColor: '#ff6347',
    padding: 15,
    borderRadius: 5,
  },
  buttonText: {
    color: '#fff',
    fontSize: 16,
    textAlign: 'center',
  },
  profileCard: {
    padding: 20,
    backgroundColor: '#fff',
    borderRadius: 10,
    elevation: 3,
  },
});
```

```
export default App;
```

Signup

**Signup**

Signup