



Assignment # 04

Mobile Application development

Sir Kamran

Muhammad Moaaz Safdar

Sp22-bse-053

24 December 2024

## Code

### App.js

```
import React, { useEffect, useState } from 'react';
import { StyleSheet, View, Text, Button, ActivityIndicator } from
'react-native';
import * as Location from 'expo-location';
import AsyncStorage from '@react-native-async-storage/async-storage';

const App = () => {
  const [location, setLocation] = useState(null);
  const [errorMsg, setErrorMsg] = useState(null);
  const [loading, setLoading] = useState(true);

  // Fetch user location
  const fetchLocation = async () => {
    try {
      // Request permission
      let { status } = await
Location.requestForegroundPermissionsAsync();
      if (status !== 'granted') {
        setErrorMsg('Permission to access location was denied');
        setLoading(false);
        return;
      }

      // Get location
      let currentLocation = await Location.getCurrentPositionAsync({});
      const coords = {
        latitude: currentLocation.coords.latitude,
        longitude: currentLocation.coords.longitude,
      };

      // Save to AsyncStorage
      await AsyncStorage.setItem('userLocation',
JSON.stringify(coords));

      // Update state
      setLocation(coords);
    } catch (error) {
      setErrorMsg('Failed to fetch location');
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
```

```
    fetchLocation();  
  }, []);
```

```
// Display saved location on component mount  
useEffect(() => {  
  const loadLocation = async () => {  
    const storedLocation = await AsyncStorage.getItem('userLocation');  
    if (storedLocation) {  
      setLocation(JSON.parse(storedLocation));  
    }  
  };  
  loadLocation();  
}, []);
```

```
return (  
  <View style={styles.container}>  
    <Text style={styles.header}>User Location</Text>  
    {loading ? (  
      <ActivityIndicator size="large" color="#1ecfd8" />  
    ) : errorMsg ? (  
      <Text style={styles.error}>{errorMsg}</Text>  
    ) : (  
      location && (  
        <View style={styles.locationContainer}>  
          <Text style={styles.text}>Latitude:  
{location.latitude}</Text>  
          <Text style={styles.text}>Longitude:  
{location.longitude}</Text>  
        </View>  
      )  
    )}  
    <Button title="Refresh Location" onPress={fetchLocation} />  
  </View>  
);  
};
```

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    justifyContent: 'center',  
    alignItems: 'center',  
    padding: 20,  
  },  
  header: {  
    fontSize: 24,  
    fontWeight: 'bold',  
    marginBottom: 20,  
  },  
  error: {
```

```

    color: 'red',
    fontSize: 16,
    marginBottom: 20,
  },
  locationContainer: {
    marginVertical: 20,
  },
  text: {
    fontSize: 16,
  },
});

```

```
export default App;
```

## LoginScreen.js

```

// LoginScreen.js
import React, { useState } from 'react';
import { View, TextInput, Button, Text, StyleSheet } from 'react-native';
import { auth } from '../firebaseConfig';
import { signInWithEmailAndPassword } from 'firebase/auth';
import * as Location from 'expo-location';
import AsyncStorage from '@react-native-async-storage/async-storage';

```

```

const LoginScreen = ({ navigation }) => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');

```

```

  const handleLogin = async () => {
    try {
      await signInWithEmailAndPassword(auth, email, password);

```

```

      // Get user's location
      const { status } = await
Location.requestForegroundPermissionsAsync();
      if (status !== 'granted') {
        setError('Permission to access location was denied');
        return;
      }
      const location = await Location.getCurrentPositionAsync({});
      const userLocation = {
        latitude: location.coords.latitude,
        longitude: location.coords.longitude,
      };

```

```

      // Store location in AsyncStorage

```

```
    await AsyncStorage.setItem('userLocation',
JSON.stringify(userLocation));
```

```
    // Navigate to Profile screen after successful Login
    navigation.navigate('Profile');
  } catch (err) {
    setError(err.message);
  }
};
```

```
return (
  <View style={styles.container}>
    <TextInput
      style={styles.input}
      placeholder="Email"
      value={email}
      onChangeText={setEmail}
    />
    <TextInput
      style={styles.input}
      placeholder="Password"
      secureTextEntry
      value={password}
      onChangeText={setPassword}
    />
    {error && <Text style={styles.error}>{error}</Text>}
    <Button title="Login" onPress={handleLogin} />
  </View>
);
};
```

```
const styles = StyleSheet.create({
  container: { padding: 20, flex: 1, justifyContent: 'center' },
  input: { height: 40, borderColor: 'gray', borderWidth: 1,
marginBottom: 10, paddingLeft: 10 },
  error: { color: 'red', marginBottom: 10 },
});
```

```
export default LoginScreen;
```

## SignUpScreen.js

```
// SignUpScreen.js
import React, { useState } from 'react';
import { View, TextInput, Button, Text, StyleSheet } from 'react-
native';
import { auth } from '../firebaseConfig';
import { createUserWithEmailAndPassword } from 'firebase/auth';
```

```
import * as Location from 'expo-location';
import AsyncStorage from '@react-native-async-storage/async-storage';
```

```
const SignupScreen = ({ navigation }) => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');
```

```
  const handleSignup = async () => {
    try {
      const userCredential = await createUserWithEmailAndPassword(auth,
email, password);
      const { user } = userCredential;
```

```
      // Get user's Location
      const { status } = await
Location.requestForegroundPermissionsAsync();
      if (status !== 'granted') {
        setError('Permission to access location was denied');
        return;
      }
      const location = await Location.getCurrentPositionAsync({});
      const userLocation = {
        latitude: location.coords.latitude,
        longitude: location.coords.longitude,
      };

```

```
      // Store Location in AsyncStorage
      await AsyncStorage.setItem('userLocation',
JSON.stringify(userLocation));
      await AsyncStorage.setItem('userEmail', email); // Optionally
store email too

```

```
      // Navigate to Profile screen after successful signup
      navigation.navigate('Profile');
    } catch (err) {
      setError(err.message);
    }
  };

```

```
  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Email"
        value={email}
        onChangeText={setEmail}
      />
      <TextInput

```

```

        style={styles.input}
        placeholder="Password"
        secureTextEntry
        value={password}
        onChangeText={setPassword}
      />
      {error && <Text style={styles.error}>{error}</Text>}
      <Button title="Sign Up" onPress={handleSignup} />
    </View>
  );
};

```

```

const styles = StyleSheet.create({
  container: { padding: 20, flex: 1, justifyContent: 'center' },
  input: { height: 40, borderColor: 'gray', borderWidth: 1,
marginBottom: 10, paddingLeft: 10 },
  error: { color: 'red', marginBottom: 10 },
});

```

```

export default SignupScreen;

```

## ProfileScreen.js

```

// ProfileScreen.js
import React, { useEffect, useState } from 'react';
import { View, Text, StyleSheet } from 'react-native';
import AsyncStorage from '@react-native-async-storage/async-storage';

```

```

const ProfileScreen = () => {
  const [userLocation, setUserLocation] = useState(null);
  const [userEmail, setUserEmail] = useState(null);

```

```

  useEffect(() => {
    const loadUserData = async () => {
      const location = await AsyncStorage.getItem('userLocation');
      const email = await AsyncStorage.getItem('userEmail');
      if (location && email) {
        setUserLocation(JSON.parse(location));
        setUserEmail(email);
      }
    };
  });

```

```

    loadUserData();
  }, []);

```

```

  return (

```

```

    <View style={styles.container}>
      {userLocation && userEmail ? (
        <>
          <Text style={styles.text}>Email: {userEmail}</Text>
          <Text style={styles.text}>Latitude:
{userLocation.latitude}</Text>
          <Text style={styles.text}>Longitude:
{userLocation.longitude}</Text>
        </>
      ) : (
        <Text style={styles.text}>No user data found.</Text>
      )}
    </View>
  );
};

```

```

const styles = StyleSheet.create({
  container: { padding: 20, flex: 1, justifyContent: 'center' },
  text: { fontSize: 18, marginBottom: 10 },
});

```

```

export default ProfileScreen;

```

## firebaseConfig.js

```

import { initializeApp } from 'firebase/app';
import { getAuth } from 'firebase/auth';
import { getFirestore } from 'firebase/firestore';

```

```

const firebaseConfig = {
  apiKey: 'YOUR_API_KEY',
  authDomain: 'YOUR_AUTH_DOMAIN',
  projectId: 'YOUR_PROJECT_ID',
  storageBucket: 'YOUR_STORAGE_BUCKET',
  messagingSenderId: 'YOUR_MESSAGING_SENDER_ID',
  appId: 'YOUR_APP_ID',
};

```

```

const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getFirestore(app);

```

```

export { auth, db };

```



OUTPUT

## User Location

Latitude: 33.7802165

Longitude: 72.3517547

REFRESH LOCATION

## Restaurant Overview



### New Town Hotel

Downtown, City Center

**4.5**

Rating

**Top 10**

Ranking

### Latest Orders

[VIEW ALL](#)



**Grilled Chicken,  
Naan, Pepsi 250ml** **PKR 750**

Nov 29, 8:20 PM



**Beef Steak,  
Garlic Bread,  
Coke 500ml** **PKR 1,200**

### Login

#### Login

[LOGIN](#)

[Don't have an account? Signup](#)

## Login

### Login

Password is required

LOGIN

Don't have an account? [Signup](#)

## Signup

### Signup

SIGNUP

Already have an account? [Login](#)

## Signup

### Signup

Password is required

SIGNUP

[Already have an account? Login](#)

## Signup

### Signup

Email is required

SIGNUP

[Already have an account? Login](#)

