



# **MOBILE APPLICATION DEVELOPMENT LAB TERMINAL**

**NAME:**

**MUHAMMAD MUA AZ SHOAIB**

**REG. NO:**

**FA20-BCS-074**

**CLASS:**

**BCS-6B**

**SUBMITTED TO:**

**MR. BABAR SHAHZAD**

**DATE:**

**15-JUNE-2023**

Create an application "My Story" with following features.

- 1- Add 4 tabs in application
  - a. Profile
  - b. My Favourites (default Screen)
  - c. My Todos
  - d. Settings
- 2- The title of each tab should be same, which will be your name. Store the username in Local Storage.

Profile:

- 1- Add a profile tab where the user can view and edit their profile information.
- 2- Display the user's name (retrieved from Local Storage) and allow them to update it if desired.
- 3- Include additional fields such as email, phone number, and a profile picture.
- 4- Allow the user to edit and save their profile information.

My Favourites

- 1- Get the posts from the link below. And display only those posts which have a userId same as your registration number, for example. FA19-bcs-010 will display all post with userId "10"

<https://jsonplaceholder.typicode.com/posts>

- 2- Under each post display the comments. (No more than 3 comments). Make sure the Id of the post matches the Post Id from step 1.

My Todos

- 1- Get the todos list from the link below. Display a maximum of 5 todos using flat list, where userId matches your registration number. If the todo is complete, background color should green else it should be red.

<https://jsonplaceholder.typicode.com/todos>

Settings

- 1- Allow user to set dark/Light theme in Settings. The selected theme should be applied on all screens and should also work when user open app in future.
- 2- Add custom theme ask user the set the background, header, text color etc.

Social Sharing:

- 1- Add social sharing functionality to allow users to share their favorite posts or their todo list with others.
- 2- Include options to share via popular social media platforms or through other communication channels, such as email or messaging apps.

## SCREENSHOTS OF APP

Muaaz Shoaib

# Sign In



Name



Email



Password



Phone

**SAVE DATA**

**LOAD DATA**



Muaaz Shoaib

# Profile

Name: Ali

Email: muaazshoaib1670@gmail.com

Password: QWERTY

Phone Number: 1234567890



Muaaz Shoaib

## Favorites

7: voluptatem doloribus  
consectetur est ut ducimus

7: beatae enim quia vel

7: voluptas blanditiis  
repellendus animi ducimus  
error sapiente et suscipit

7: et fugit quas eum in in  
aperiam quod

7: consequatur id enim sunt et  
et

7: repudiandae ea animi iusto

7: aliquid eos sed fuga est



Muaaz Shoaib

# Todo

1: delectus aut autem

1: quis ut nam facilis et  
officia qui

1: fugiat veniam minus

1: et porro tempora

1: laboriosam mollitia et  
enim quasi adipisci quia  
provident illum



Muaaz Shoaib

---

## Change Theme



Ali

---

## Sign In



Name



Email



Password



Phone

**SAVE DATA**

**LOAD DATA**





## APP.JS

```
import * as React from "react";
import { useState, useEffect, useContext } from "react";
import AsyncStorage from "@react-native-async-storage/async-storage";

import { NavigationContainer } from "@react-navigation/native";
import { EventRegister } from "react-native-event-listeners";
import TabNavigation from "../navigation/TabNavigation";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import themeContext from "../config/themeContext";
import theme from "../config/theme";

const Stack = createNativeStackNavigator();
export default function App() {
  const [mode, setMode] = useState(false);
  const [title, setTitle] = useState("");

  useEffect(() => {
    const fetchStackTitle = async () => {
      try {
        var userInformation = await AsyncStorage.getItem("@storage_Key");
        userInformation = JSON.parse(userInformation);
        setTitle(userInformation["name"]);
      } catch (error) {
        console.log("Error retrieving stack title:", error);
      }
    };

    fetchStackTitle();
  }, []);

  useEffect(() => {
    let eventListener = EventRegister.addEventListener(
      "changeTheme",
      (data) => {
        setMode(data);
        console.log(data);
      }
    );
    return () => {
      EventRegister.removeEventListener(eventListener);
    };
  });
});
```

```

return (
  <themeContext.Provider value={mode === true ? theme.dark : theme.light}>
    <NavigationContainer>
      <Stack.Navigator
        screenOptions={{
          // headerShown: false,
          title: title,
        }}
      >
        <Stack.Screen
          name="TabNavigation"
          component={TabNavigation}
        ></Stack.Screen>
      </Stack.Navigator>
    </NavigationContainer>
  </themeContext.Provider>
);
}

```

## TABNAVIGATION.JS

```

import * as React from "react";

import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";

import SignIn from "../pages/SignIn";
import Profile from "../pages/Profile";
import Favorites from "../pages/Favorites";
import Todo from "../pages/Todo";
import Settings from "../pages/Settings";
import { Ionicons } from "@expo/vector-icons";

export default function TabNavigation() {
  const Tab = createBottomTabNavigator();
  return (
    <Tab.Navigator
      screenOptions={({ route }) => ({
        tabBarIcon: ({ focused, color, size }) => {
          let iconName;
          <ion-icon name="log-in-outline"></ion-icon>;
          if (route.name === "SignIn") {
            iconName = focused ? "log-in" : "log-in-outline";
          } else if (route.name === "Profile") {

```

```

        iconName = focused ? "person" : "person-outline";
    } else if (route.name === "Favorites") {
        iconName = focused ? "heart" : "heart-outline";
    } else if (route.name === "Todo") {
        iconName = focused
            ? "checkmark-circle"
            : "checkmark-circle-outline";
    } else if (route.name === "Settings") {
        iconName = focused ? "settings" : "settings-outline";
    }

    return <Ionicons name={iconName} size={size} color={color} />;
},
tabBarActiveTintColor: "blue",
tabBarInactiveTintColor: "gray",
headerShown: false,
tabBarShowLabel: false,
}}}
>
<Tab.Screen name="SignIn" component={SignIn} />
<Tab.Screen name="Profile" component={Profile} />
<Tab.Screen name="Favorites" component={Favorites} />
<Tab.Screen name="Todo" component={Todo} />
<Tab.Screen name="Settings" component={Settings} />
</Tab.Navigator>
);
}

```

## THEME.JS

```

const theme = {
  light: {
    theme: "light",
    color: "black",
    background: "white",
  },
  dark: {
    theme: "dark",
    color: "white",
    background: "black",
  },
};

export default theme;

```

## THEMECONTEXT.JS

```
import React, { createContext } from "react";

const themeContext = createContext({});

export default themeContext;
```

## SIGNIN.JS

```
import * as React from "react";
import {
  Text,
  View,
  StyleSheet,
  TouchableOpacity,
  TextInput,
} from "react-native";
import { useState } from "react";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import { Feather } from "@expo/vector-icons";
import AsyncStorage from "@react-native-async-storage/async-storage";
import themeContext from "../config/themeContext";
import { useContext } from "react";

const Stack = createNativeStackNavigator();

const Page = ({ navigation, route }) => {
  const theme = useContext(themeContext);

  const [email, setEmail] = useState(null);
  const [name, setName] = useState(null);
  const [password, setPassword] = useState(null);
  const [phoneNumber, setPhoneNumber] = useState(null);

  return (
    <View style={[styles.container, { backgroundColor: theme.background }]}>
      <Text style={[styles.title, { color: theme.color }]}>Sign In</Text>

      <View style={styles.inputContainer}>
        <Feather name="user" size={24} color="#999" />
        <TextInput
          style={styles.input}
          placeholder="Name"
        />
      </View>
    </View>
  );
};
```

```

        placeholderTextColor="#999"
        value={name}
        onChangeText={(text) => setName(text)}
    />
</View>
<View style={styles.inputContainer}>
    <Feather name="mail" size={24} color="#999" />
    <TextInput
        style={styles.input}
        placeholder="Email"
        placeholderTextColor="#999"
        value={email}
        onChangeText={(text) => setEmail(text)}
    />
</View>
<View style={styles.inputContainer}>
    <Feather name="lock" size={24} color="#999" />
    <TextInput
        style={styles.input}
        placeholder="Password"
        placeholderTextColor="#999"
        secureTextEntry
        value={password}
        onChangeText={(text) => setPassword(text)}
    />
</View>

<View style={styles.inputContainer}>
    <Feather name="phone" size={24} color="#999" />
    <TextInput
        style={styles.input}
        placeholder="Phone"
        placeholderTextColor="#999"
        value={phoneNumber}
        onChangeText={(text) => setPhoneNumber(text)}
    />
</View>

<TouchableOpacity
    style={styles.button}
    onPress={async () => {
        if (
            name == null ||
            email == null ||
            password == null ||

```

```

        phoneNumber == null
    ) {
        alert("Please fill out all fields.");
        return;
    }

    try {
        const userInformation = {
            name: name,
            email: email,
            password: password,
            phoneNumber: phoneNumber,
        };
        await AsyncStorage.setItem(
            "@storage_Key",
            JSON.stringify(userInformation)
        );
        console.log("Data is saved!");
        console.log(userInformation);
        alert("Data is saved!");
    } catch (e) {}
  }}
>
  <Text style={styles.buttonText}>SAVE DATA</Text>
</TouchableOpacity>

<TouchableOpacity
  style={styles.button}
  onPress={async () => {
    try {
      var userInformation = await AsyncStorage.getItem("@storage_Key");
      userInformation = JSON.parse(userInformation);

      navigation.navigate("Profile", {
        Name: userInformation["name"],
        Email: userInformation["email"],
        Password: userInformation["password"],
        PhoneNumber: userInformation["phoneNumber"],
      });
    } catch (e) {}
  }}
>
  <Text style={styles.buttonText}>LOAD DATA</Text>
</TouchableOpacity>
</View>

```

```

    );
  };

const SignIn = ({ navigation }) => {
  return (
    <Stack.Navigator
      screenOptions={{
        headerShown: false,
      }}
    >
      <Stack.Screen name="SignIn" component={Page}></Stack.Screen>
    </Stack.Navigator>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
    backgroundColor: "#F5F5F5",
  },
  inputContainer: {
    flexDirection: "row",
    alignItems: "center",
    width: "80%",
    height: 48,
    borderRadius: 5,
    backgroundColor: "#FFF",
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 3.84,
    elevation: 5,
    paddingLeft: 16,
    marginBottom: 16,
  },
  title: {
    fontSize: 32,
    fontWeight: "bold",
    marginTop: 24,
    marginBottom: 24,
  },
},

```

```

input: {
  flex: 1,
  marginLeft: 16,
  fontSize: 16,
},
button: {
  width: "80%",
  height: 48,
  borderRadius: 5,
  backgroundColor: "#007AFF",
  alignItems: "center",
  justifyContent: "center",
  marginTop: 10,
},

buttonText: {
  color: "#FFF",
  fontSize: 16,
  fontWeight: "bold",
},
});
export default SignIn;

```

## PROFILE.JS

```

import * as React from "react";
import { Text, View, StyleSheet, TouchableOpacity } from "react-native";
import themeContext from "../config/themeContext";
import { useContext } from 'react';

const Profile = ({ route }) => {
  const theme = useContext(themeContext);
  const { Name, Email, Password, PhoneNumber } = route.params;
  return (
    <View style={[styles.container, { backgroundColor: theme.background }]}>
      <Text style={[styles.title, { color: theme.color }]}>Profile</Text>

      <TouchableOpacity style={styles.button}>
        <Text style={styles.buttonText}>Name: {Name}</Text>
      </TouchableOpacity>

      <TouchableOpacity style={styles.button}>
        <Text style={styles.buttonText}>Email: {Email}</Text>

```



```

    </TouchableOpacity>

    <TouchableOpacity style={styles.button}>
      <Text style={styles.buttonText}>Password: {Password}</Text>
    </TouchableOpacity>

    <TouchableOpacity style={styles.button}>
      <Text style={styles.buttonText}>Phone Number: {PhoneNumber}</Text>
    </TouchableOpacity>
  </View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
    backgroundColor: "#F5F5F5",
  },

  title: {
    fontSize: 32,
    fontWeight: "bold",
    marginTop: 24,
    marginBottom: 24,
  },

  button: {
    width: "80%",
    height: 48,
    borderRadius: 5,
    backgroundColor: "#007AFF",
    alignItems: "center",
    justifyContent: "center",
    marginTop: 15,
  },

  buttonText: {
    color: "#FFF",
    fontSize: 16,
    fontWeight: "bold",
  },
});

```

```
export default Profile;
```

## FAVORITES.JS

```
import React, { useEffect, useState } from "react";
import {
  StyleSheet,
  FlatList,
  Text,
  View,
  TouchableOpacity,
  Share,
} from "react-native";

import themeContext from "../config/themeContext";
import { useContext } from "react";

const handleShare = (message, url) => {
  Share.share({
    message: message,
    url: url,
    title: "Share Content",
  })
    .then((result) => console.log(result))
    .catch((error) => console.log(error));
};

const Favorites = () => {
  const [data, setData] = useState([]);
  const theme = useContext(themeContext);

  useEffect(() => {
    fetchData();
  }, []);

  const fetchData = async () => {
    try {
      const response = await fetch(
        "https://jsonplaceholder.typicode.com/posts"
      );
      const jsonData = await response.json();
      const filteredData = jsonData.filter((item) => item.userId === 7);
      setData(filteredData);
    }
  };
};
```

```

    } catch (error) {
      console.error("Error fetching data:", error);
    }
  };

const renderItem = ({ item }) => (
  <View style={[styles.container, { backgroundColor: theme.background }]}>
    <TouchableOpacity
      onPress={handleShare(
        item.title,
        "https://jsonplaceholder.typicode.com/posts"
      )}
      style={styles.button}
    >
      <Text style={styles.buttonText}>
        {item.userId}: {item.title}
      </Text>
    </TouchableOpacity>
  </View>
);

return (
  <View style={[styles.container, { backgroundColor: theme.background }]}>
    <Text style={[styles.title, { color: theme.color }]}>Favorites</Text>

    <FlatList
      data={data}
      renderItem={renderItem}
      keyExtractor={(item) => item.id.toString()}
    />
  </View>
);
};

export default Favorites;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
    backgroundColor: "#F5F5F5",
  },
  title: {

```

```

    fontSize: 32,
    fontWeight: "bold",
    marginTop: 50,
    marginBottom: 24,
  },

  button: {
    width: "80%",
    height: 60,
    borderRadius: 5,
    backgroundColor: "#007AFF",
    alignItems: "center",
    justifyContent: "center",
    marginTop: 15,
  },

  buttonText: {
    color: "#FFF",
    fontSize: 16,
    fontWeight: "bold",
  },
});

```

## TODO.JS

```

import React, { useEffect, useState } from "react";
import {
  StyleSheet,
  FlatList,
  Text,
  View,
  TouchableOpacity,
  Share,
} from "react-native";

import themeContext from "../config/themeContext";
import { useContext } from "react";

const handleShare = (message, url) => {
  Share.share({
    message: message,
    url: url,
    title: "Share Content",
  });

```

```

    })
    .then((result) => console.log(result))
    .catch((error) => console.log(error));
  });

const Todo = () => {
  const theme = useContext(themeContext);

  const [data, setData] = useState([]);

  useEffect(() => {
    fetchData();
  }, []);

  const fetchData = async () => {
    try {
      const response = await fetch(
        "https://jsonplaceholder.typicode.com/todos"
      );
      const jsonData = await response.json();
      setData(jsonData.slice(0, 5));
    } catch (error) {
      console.error("Error fetching data:", error);
    }
  };

  const renderItem = ({ item }) => (
    <View style={[styles.container, { backgroundColor: theme.background }]}>
      <TouchableOpacity
        onPress={handleShare(
          item.title,
          "https://jsonplaceholder.typicode.com/todos"
        )}>
        <Text style={{
          width: "80%",
          height: 60,
          borderRadius: 5,
          backgroundColor: item.completed ? "green" : "red",
          alignItems: "center",
          justifyContent: "center",
          marginTop: 15,
        }}>
          <Text style={styles.buttonText}>
            {item.userId}: {item.title}
          </Text>
        </Text>
      </TouchableOpacity>
    </View>
  );

```

```

        </Text>
      </TouchableOpacity>
    </View>
  );

  return (
    <View style={[styles.container, { backgroundColor: theme.background }]}>
      <Text style={[styles.title, { color: theme.color }]}>Todo</Text>
      <FlatList
        data={data}
        renderItem={renderItem}
        keyExtractor={(item) => item.id.toString()}
      />
    </View>
  );
};

export default Todo;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
  },

  title: {
    fontSize: 32,
    fontWeight: "bold",
    marginTop: 50,
    marginBottom: 24,
  },

  button: {
    width: "80%",
    height: 48,
    borderRadius: 5,
    alignItems: "center",
    justifyContent: "center",
    marginTop: 15,
  },

  buttonText: {
    color: "#FFF",
    fontSize: 16,
  },
});

```

```
    fontWeight: "bold",
  },
});
```

## SETTINGS.JS

```
import React, { useContext, useState } from "react";
import { Text, View, StyleSheet, Switch } from "react-native";

import { EventRegister } from "react-native-event-listeners";
import themeContext from "../config/themeContext";

const Settings = () => {
  const theme = useContext(themeContext);
  const [mode, setMode] = useState(false);
  return (
    <View style={[styles.container, { backgroundColor: theme.background }]}>
      <Text style={[styles.title, { color: theme.color }]}>Change Theme</Text>
      <Switch
        value={mode}
        onChange={(value) => {
          setMode(value);
          EventRegister.emit("changeTheme", value);
        }}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
    backgroundColor: "#F5F5F5",
  },

  title: {
    fontSize: 32,
    fontWeight: "bold",
    marginTop: 24,
    marginBottom: 24,
  },
});
```

```
button: {
  width: "80%",
  height: 48,
  borderRadius: 5,
  backgroundColor: "#007AFF",
  alignItems: "center",
  justifyContent: "center",
  marginTop: 15,
},

buttonText: {
  color: "#FFF",
  fontSize: 16,
  fontWeight: "bold",
},
});

export default Settings;
```