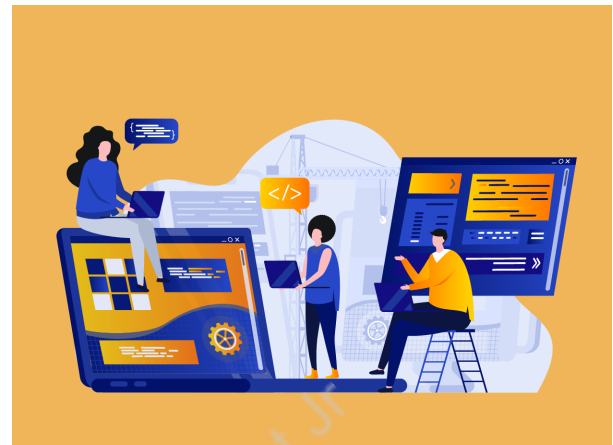


## THEMES



### What is our GOAL for this MODULE?

In this class, we learned to integrate the light theme into our storytelling app, so that the user's can choose between the light and dark themes as per their preference.

### What did we ACHIEVE in the class TODAY?

- Created a light theme for our app.
- Integrated themes into the app so that users can choose the preferred one.

### Which CONCEPTS/ CODING BLOCKS did we cover today?

- Changing themes using a toggle switch.

## How did we DO the activities?

1. Head to the **firebase** database and change one of our user's themes to light for testing.



2. Create a state called **light\_theme** in our profile screen, which is true if the user has preferred light theme and false, if the user has preferred the dark theme.

Note: The changes are highlighted for reference purposes in the code snippets.

```
return (
  <View style={this.state.light_theme ? styles.containerLight :
styles.container}>
    <SafeAreaView style={styles.droidSafeArea} />
    <View style={styles.appTitle}>
      <View style={styles.appIcon}>
        <Image source={require("../assets/logo.png")}
style={{ width: 60, height: 60, resizeMode: 'contain', marginLeft: 10
}}></Image>
      </View>
      <View style={styles.appTitleTextContainer}>
        <Text style={this.state.light_theme ?
styles.appTitleTextLight : styles.appTitleText}>
          Story Telling App
        </Text>
      </View>
    </View>
  </View>
  <View style={styles.screenContainer}>
```

```

    <View style={styles.profileImageContainer}>
      <Image source={{ uri: this.state.profile_image }}
style={styles.profileImage}></Image>
    </View>
    <View style={styles.nameContainer}>
      <Text style={this.state.light_theme ?
styles.nameTextLight : styles.nameText}>{this.state.name}</Text>
    </View>
    <View style={styles.themeContainer}>
      <View style={styles.themeTextContainer}>
        <Text style={this.state.light_theme ?
styles.themeTextLight : styles.themeText}>Dark Theme</Text>
      </View>
      <View style={styles.switchContainer}>
        <Switch
          style={{ transform: [{ scaleX: 1.3 }, {
scaleY: 1.3 }] }}
          trackColor={{ false: "#767577", true:
this.state.light_theme ? "#eee" : "white" }}
          thumbColor={this.state.isEnabled ? "#ee8249"
: "#f4f3f4"}
          ios_backgroundColor="#3e3e3e"
          onChange={() => this.toggleSwitch()}
          value={this.state.isEnabled}
        />
      </View>
    </View>
  </View>
</View>

```

3. Add styles to all the components declared in the **Profile.js** screen file.

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#15193c"
  },
  containerLight: {
    flex: 1,
    backgroundColor: "white"
  }
})

```

```
},
droidSafeArea: {
  marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 0
},
appTitle: {
  flex: 0.07,
  flexDirection: "row",
  flexWrap: "wrap",
  padding: 5,
},
appIcon: {
  flex: 0.3
},
appTitleTextContainer: {
  justifyContent: "center",
  alignItems: "center"
},
appTitleText: {
  color: "white",
  fontSize: 28,
  fontFamily: "Bubblegum-Sans",
  paddingLeft: 20
},
appTitleTextLight: {
  color: "black",
  fontSize: 28,
  fontFamily: "Bubblegum-Sans",
  paddingLeft: 20
},
screenContainer: {
  flex: 0.85
},
profileImageContainer: {
  flex: 0.3,
  marginTop: 50,
  alignItems: "center"
},
profileImage: {
  width: 150,
  height: 150,
  borderRadius: 150 / 2,
```

```
    },
    nameContainer: {
      flex: 0.1,
      alignItems: "center"
    },
    nameText: {
      color: "white",
      fontSize: 40,
      fontFamily: "Bubblegum-Sans",
    },
    nameTextLight: {
      color: "black",
      fontSize: 40,
      fontFamily: "Bubblegum-Sans",
    },
    themeContainer: {
      flexDirection: "row",
      justifyContent: "center",
      paddingTop: 80
    },
    themeTextContainer: {
      alignItems: "center",
      flex: 0.5
    },
    themeText: {
      color: "white",
      fontSize: 30,
      fontFamily: "Bubblegum-Sans",
    },
    themeTextLight: {
      color: "black",
      fontSize: 30,
      fontFamily: "Bubblegum-Sans",
    },
    switchContainer: {
      justifyContent: "center",
      alignItems: "center"
    }
  }
});
```

4. Fetch the user to get their preferred theme in **FeedScreen.js**.

- Import **firebase**.

```
import firebase from "firebase";
```

- Add an extra state **light\_theme** to our constructor.

```
constructor(props) {
  super(props);
  this.state = {
    fontsLoaded: false,
    light_theme: true
  };
}
```

- Create the **fetchUser()** function -

```
fetchUser = () => {
  let theme;
  firebase
    .database()
    .ref("/users/" + firebase.auth().currentUser.uid)
    .on("value", (snapshot) => {
      theme = snapshot.val().current_theme
      this.setState({ light_theme: theme === "light" })
    })
}
```

- Call **fetchUser()** within the **componentDidMount()** function.

```
componentDidMount() {
  this._loadFontsAsync();
  this.fetchUser();
}
```

- Update the **render()** function of the **FeedScreen.js** file.

```
render() {
```

```

    if (!this.state.fontsLoaded) {
      return <AppLoading />;
    } else {
      return (
        <View style={this.state.light_theme ?
styles.containerLight : styles.container}>
          <SafeAreaView style={styles.droidSafeArea} />
          <View style={styles.appTitle}>
            <View style={styles.appIcon}>
              <Image source={require("../assets/logo.png")}
style={{ width: 60, height: 60, resizeMode: 'contain', marginLeft: 10
}}></Image>
            </View>
            <View style={styles.appTitleTextContainer}>
              <Text style={this.state.light_theme ?
styles.appTitleTextLight : styles.appTitleText}>
                Storytelling App
              </Text>
            </View>
          </View>
          <View style={styles.cardContainer}>
            <FlatList
              keyExtractor={this.keyExtractor}
              data={stories}
              renderItem={this.renderItem}
            />
          </View>
        </View>
      )
    }
  }
}

```

Now change the styling for the **StoryCard** and our **Bottom Tab Navigator** by updating the **StoryCard** first.

5. Import **firebase**, add **light\_theme** state to the **constructor()** and create a **fetchUser()** function and call it in **componentDidMount()** (similar to step 4) in **StoryCard**.

```

return (
  <TouchableOpacity style={styles.container} onPress={() =>
this.props.navigation.navigate("StoryScreen", story = this.props.story)}>
    <SafeAreaView style={styles.droidSafeArea} />
  </TouchableOpacity>
)

```

```

<View style={this.state.light_theme ?
styles.cardContainerLight : styles.cardContainer}>
  <View style={styles.storyImage}>
    <Image
      source={require("../assets/story_image_1.png")} style={{ resizeMode: 'contain',
width: Dimensions.get('window').width - 60, height: 250, borderRadius: 10
}}></Image>
  </View>
  <View style={styles.titleContainer}>
    <View style={styles.titleTextContainer}>
      <View style={styles.storyTitle}>
        <Text style={this.state.light_theme ?
styles.storyTitleTextLight :
styles.storyTitleText}>{this.props.story.title}</Text>
      </View>
      <View style={styles.storyAuthor}>
        <Text style={this.state.light_theme ?
styles.storyAuthorTextLight :
styles.storyAuthorText}>{this.props.story.author}</Text>
      </View>
    </View>
  </View>
  <View style={styles.descriptionContainer}>
    <Text style={this.state.light_theme ?
styles.descriptionTextLight : styles.descriptionText}>
      {this.props.story.description}
    </Text>
  </View>
  <View style={styles.actionContainer}>
    <View style={styles.likeButton}>
      <View style={styles.likeIcon}>
        <Icons name={"heart"} size={30}
color={this.state.light_theme ? "black" : "white"} style={{ width: 30,
marginLeft: 20, marginTop: 5 }} />
      </View>
      <View>
        <Text style={this.state.light_theme ?
styles.likeTextLight : styles.likeText}>12k</Text>
      </View>
    </View>
  </View>
</View>

```



```

    </View>
  </TouchableOpacity>
)

```

And the styles would be as follows -

```

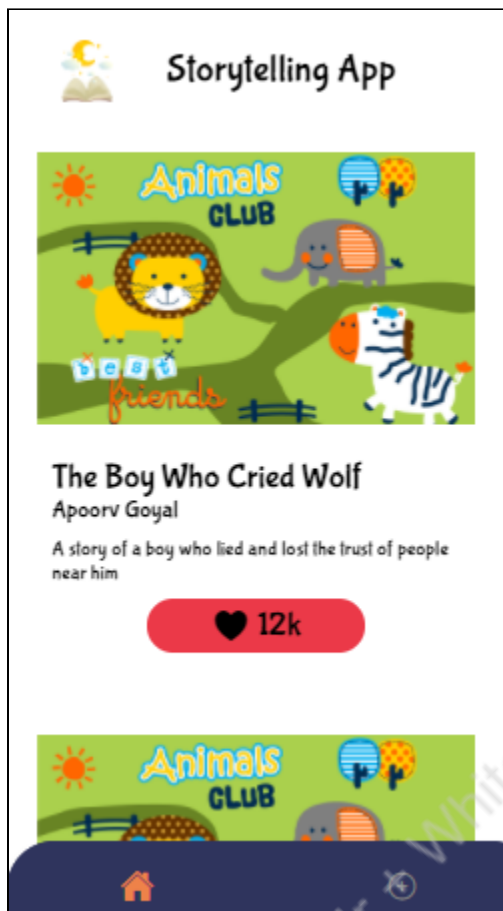
const styles = StyleSheet.create({
  droidSafeArea: {
    marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 0
  },
  cardContainer: {
    marginTop: -20,
    marginBottom: 20,
    marginLeft: 20,
    marginRight: 20,
    backgroundColor: "#2f345d",
    borderRadius: 20,
    height: undefined,
    padding: 10
  },
  cardContainerLight: {
    marginTop: -20,
    marginBottom: 20,
    marginLeft: 20,
    marginRight: 20,
    backgroundColor: "white",
    borderRadius: 20,
    height: undefined,
    padding: 10,
    shadowColor: 'rgb(0, 0, 0)',
    shadowOffset: {
      width: 3,
      height: 3,
    },
    shadowOpacity: 0.5,
    shadowRadius: 5,
    elevation: 2,
  },
  titleContainer: {
    flexDirection: "row"
  },
  titleTextContainer: {

```

```
        flex: 0.8
    },
    iconContainer: {
        flex: 0.2
    },
    storyTitleText: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 25,
        color: "white"
    },
    storyTitleTextLight: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 25,
        color: "black"
    },
    storyAuthorText: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 18,
        color: "white"
    },
    storyAuthorTextLight: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 18,
        color: "black"
    },
    descriptionContainer: {
        marginTop: 5
    },
    descriptionText: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 13,
        color: "white"
    },
    descriptionTextLight: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 13,
        color: "black"
    },
    actionContainer: {
        marginTop: 10,
        justifyContent: "center",
```

```
        alignItems: "center"
      },
      likeButton: {
        backgroundColor: "#eb3948",
        borderRadius: 30,
        width: 160,
        height: 40,
        flexDirection: "row"
      },
      likeText: {
        color: "white",
        fontFamily: "Bubblegum-Sans",
        fontSize: 25,
        marginLeft: 25,
        marginTop: 6
      },
      likeTextLight: {
        fontFamily: "Bubblegum-Sans",
        fontSize: 25,
        marginLeft: 25,
        marginTop: 6
      }
    }
  });
```

Output:



6. Now, format the **Bottom Tab Navigator** as given below:
  - Import the **firebase** module as -  
`import firebase from "firebase";`
  - Change this code line from -  
`const BottomTabNavigator = () =>{`
  - To -  
`export default class BottomTabNavigator extends component{`
  - Next, **import {Component}** from "react" as follows -  
`import React, {Component} from 'react';`
7. Add our **constructor()** and **componentDidMount()** function in the **BottomTabNavigator** file.

```
constructor(props) {  
  super(props);  
  this.state = {  
    light_theme: true  
  };  
}  
  
componentDidMount() {  
  let theme;  
  firebase  
    .database()  
    .ref("/users/" + firebase.auth().currentUser.uid)  
    .on("value", function (snapshot) {  
      theme = snapshot.val().current_theme  
    })  
  this.setState({ light_theme: theme === "light" ? true : false })  
}
```

8. Wrap our **return()** function within a **render()** function -

```

render() {
  return (
    <Tab.Navigator
      labeled={false}
      barStyle={styles.bottomTabStyle}
      screenOptions={({ route }) => ({
        tabBarIcon: ({ focused, color, size }) => {
          let iconName;
          if (route.name === 'Feed') {
            iconName = focused
              ? 'home'
              : 'home-outline';
          } else if (route.name === 'Create Story') {
            iconName = focused ? 'add-circle' : 'add-circle-outline';
          }
          return <Ionicons name={iconName} size={30} color={color} style={{ w
        },
      })}
      activeColor={'#ee8249'}
      inactiveColor={'gray'}
    >
    <Tab.Screen name="Feed" component={Feed} />
    <Tab.Screen name="Create Story" component={CreateStory} />
  </Tab.Navigator>
);
}

```

9. Add styling conditions to the **barStyle** attribute as follows:

```
barStyle={this.state.light_theme ? styles.bottomTabStyleLight : styles.bottomTabStyle}
```

10. Add styles to **barStyle** props of **Tab.Navigator**

```

bottomTabStyleLight: {
  backgroundColor: "#eaeaea",
  height: "8%",
  borderTopLeftRadius: 30,
  borderTopRightRadius: 30,
  overflow: 'hidden',
  position: 'absolute'
}

```

11. Remove the export statement towards the end, since we are already exporting our newly converted class component above.

Output:



### What's next?

Next class, we will use firebase to add stories and get stories, instead of using temporary data. We will also run to check if there are any bugs in our App and debug those.

### EXTEND YOUR KNOWLEDGE

Bookmark the following link to know more about ToggleSwitch:

<https://www.npmjs.com/package/toggle-switch-react-native>