PRO-C184

FILTERS CATEGORY

Accessories
Specs
Animal Faces

What is our GOAL for this MODULE?

We learned to add different categories of the filters based on the different frames options

What did we ACHIEVE in the class TODAY?

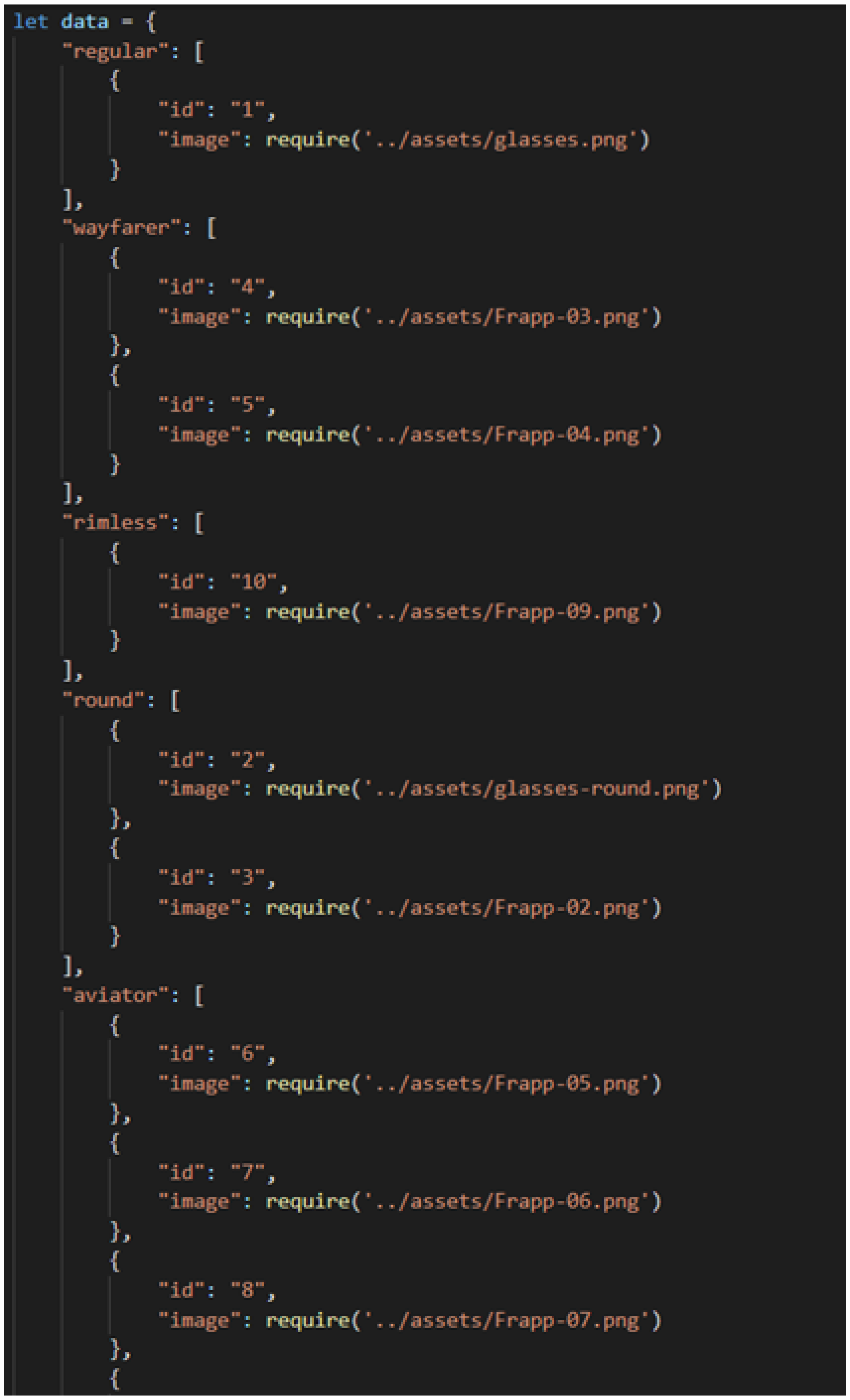
- We learned to add categories for multiple face filters on the face

Which CONCEPTS/CODING BLOCKS did we cover today?

- expo-permissions
- expo-camera
- expo-face-detector
- <Camera/>, <SafeAreaView/>,<ScrollView/>,<Platform/> components
- react-native-responsive-fontsize library
- RFPercentage, RFValue

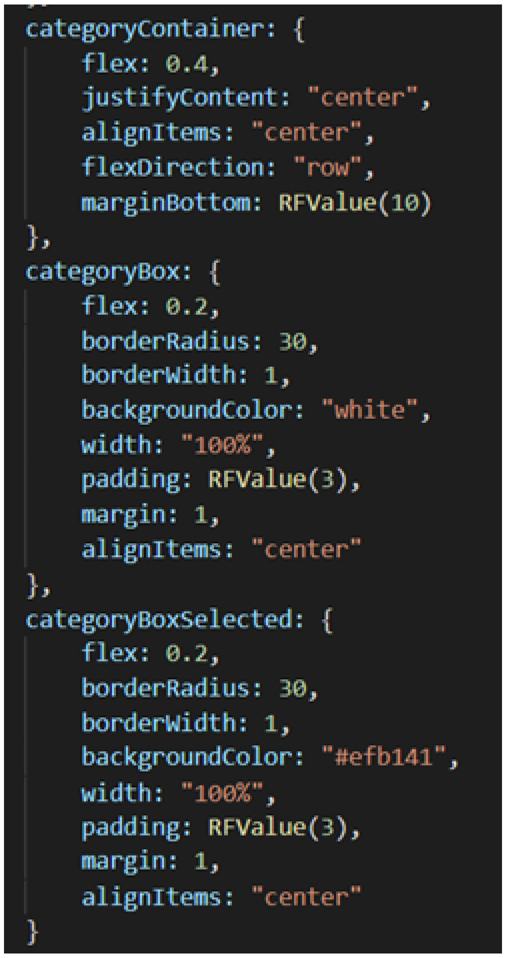
How did we DO the activities?

1. Update data variables with the categories



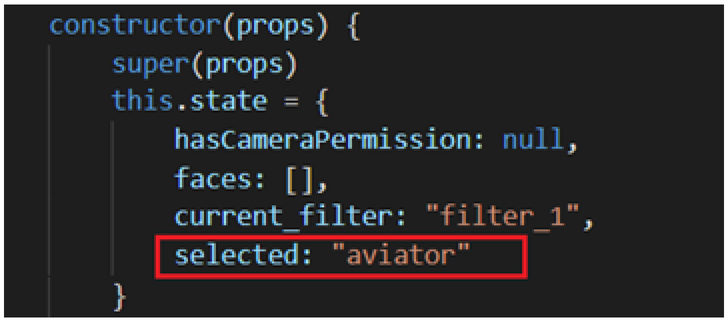
```
let data = {
    "regular": [
        {
            "id": "1",
            "image": require('../assets/glasses.png')
        }
    ],
    "wayfarer": [
        {
            "id": "4",
            "image": require('../assets/Frapp-03.png')
        },
        {
            "id": "5",
            "image": require('../assets/Frapp-04.png')
        }
    ],
    "rimless": [
        {
            "id": "10",
            "image": require('../assets/Frapp-09.png')
        }
    ],
    "round": [
        {
            "id": "2",
            "image": require('../assets/glasses-round.png')
        },
        {
            "id": "3",
            "image": require('../assets/Frapp-02.png')
        }
    ],
    "aviator": [
        {
            "id": "6",
            "image": require('../assets/Frapp-05.png')
        },
        {
            "id": "7",
            "image": require('../assets/Frapp-06.png')
        },
        {
            "id": "8",
            "image": require('../assets/Frapp-07.png')
        },
        {
```

2. Add the style for each container:
    - Category Container
    - Category Box
    - Category Box Selected



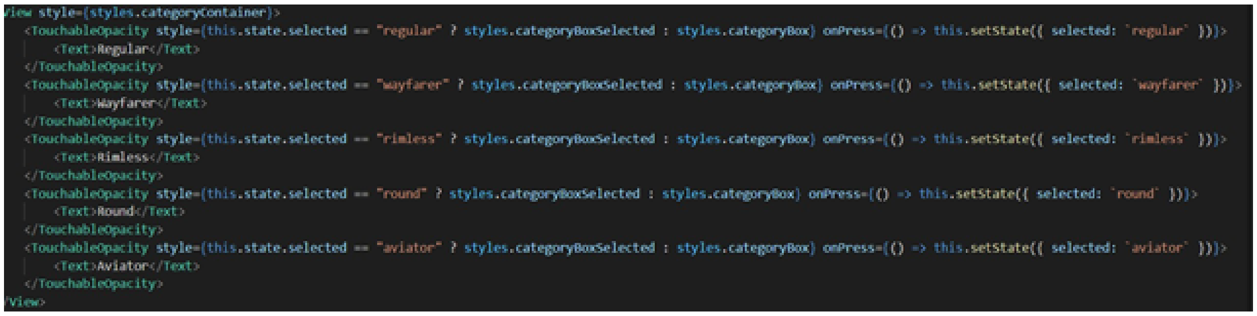
```
categoryContainer: {
    flex: 0.4,
    justifyContent: "center",
    alignItems: "center",
    flexDirection: "row",
    marginBottom: RFValue(10)
},
categoryBox: {
    flex: 0.2,
    borderRadius: 30,
    borderWidth: 1,
    backgroundColor: "white",
    width: "100%",
    padding: RFValue(3),
    margin: 1,
    alignItems: "center"
},
categoryBoxSelected: {
    flex: 0.2,
    borderRadius: 30,
    borderWidth: 1,
    backgroundColor: "#efb141",
    width: "100%",
    padding: RFValue(3),
    margin: 1,
    alignItems: "center"
}
```

3. Define the “**selected**” state variable.



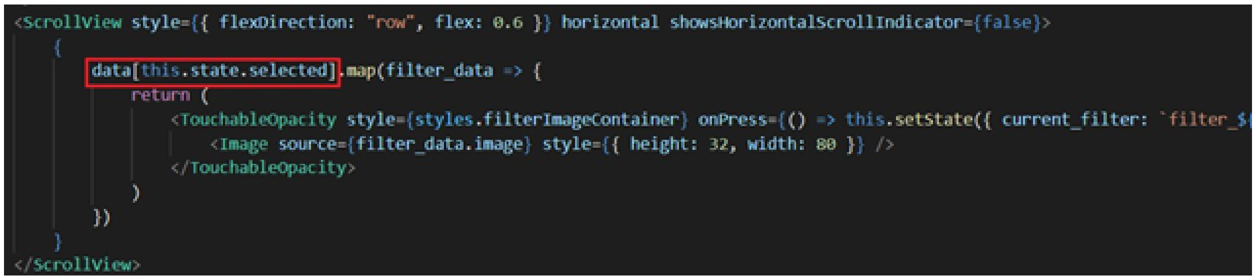
```
constructor(props) {
    super(props)
    this.state = {
        hasCameraPermission: null,
        faces: [],
        current_filter: "filter_1",
        selected: "aviator"
    }
```

4. Write a return method to render text and category boxes and update the “selected” state value.



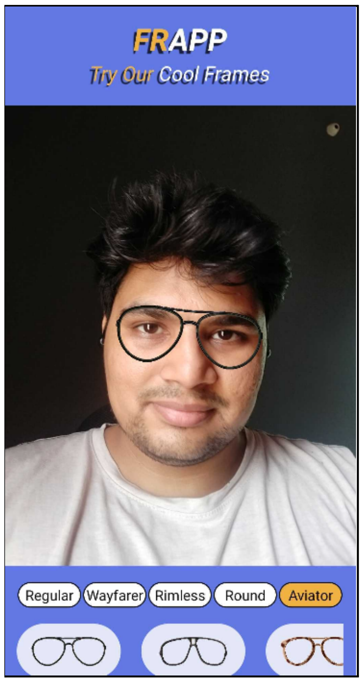
```
iew style={styles.categoryContainer}>
  <TouchableOpacity style={this.state.selected == "regular" ? styles.categoryBoxSelected : styles.categoryBox} onPress={() => this.setState({ selected: `regular` })}>
      <Text>Regular</Text>
  </TouchableOpacity>
  <TouchableOpacity style={this.state.selected == "wayfarer" ? styles.categoryBoxSelected : styles.categoryBox} onPress={() => this.setState({ selected: `wayfarer` })}>
      <Text>Wayfarer</Text>
  </TouchableOpacity>
  <TouchableOpacity style={this.state.selected == "rimless" ? styles.categoryBoxSelected : styles.categoryBox} onPress={() => this.setState({ selected: `rimless` })}>
      <Text>Rimless</Text>
  </TouchableOpacity>
  <TouchableOpacity style={this.state.selected == "round" ? styles.categoryBoxSelected : styles.categoryBox} onPress={() => this.setState({ selected: `round` })}>
      <Text>Round</Text>
  </TouchableOpacity>
  <TouchableOpacity style={this.state.selected == "aviator" ? styles.categoryBoxSelected : styles.categoryBox} onPress={() => this.setState({ selected: `aviator` })}>
      <Text>Aviator</Text>
  </TouchableOpacity>
/View>
```

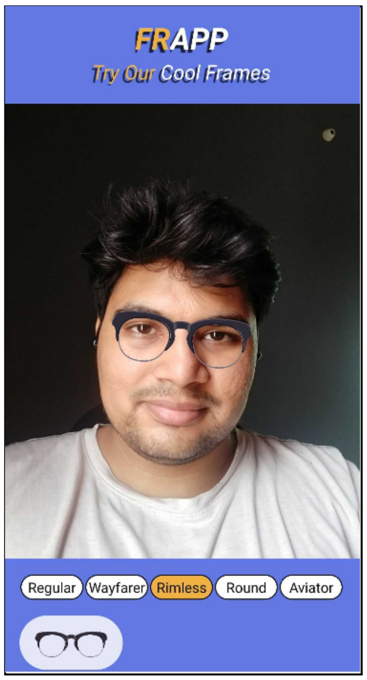
5. Render based on the state variable value.

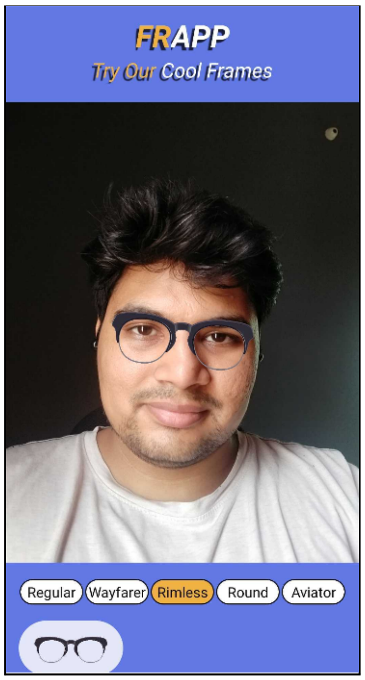


```
<ScrollView style={{ flexDirection: "row", flex: 0.6 }} horizontal showsHorizontalScrollIndicator={false}>
    {
        data[this.state.selected].map(filter_data => {
            return (
                <TouchableOpacity style={styles.filterImageContainer} onPress={() => this.setState({ current_filter: `filter_${
                    <Image source={filter_data.image} style={{ height: 32, width: 80 }} />
                </TouchableOpacity>
            )
        })
    }
</ScrollView>
```

6. Test the output by selecting different category to choose frames:







We have successfully learned to apply face filters using the data received from the FaceDetector API expo module.

What's NEXT?

In the next class, we will learn to switch between multiple filters to apply to the face.

EXTEND YOUR KNOWLEDGE:

- You can refer to the link below to explore more about expo FaceDetector
  FaceDetector