# PRO-C181

WhiteHat Jr
Live Online Coding for Kids

## FACE DETECTION


### What is our GOAL for this MODULE?

We learned about face detection to build a face filters app using react native.

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

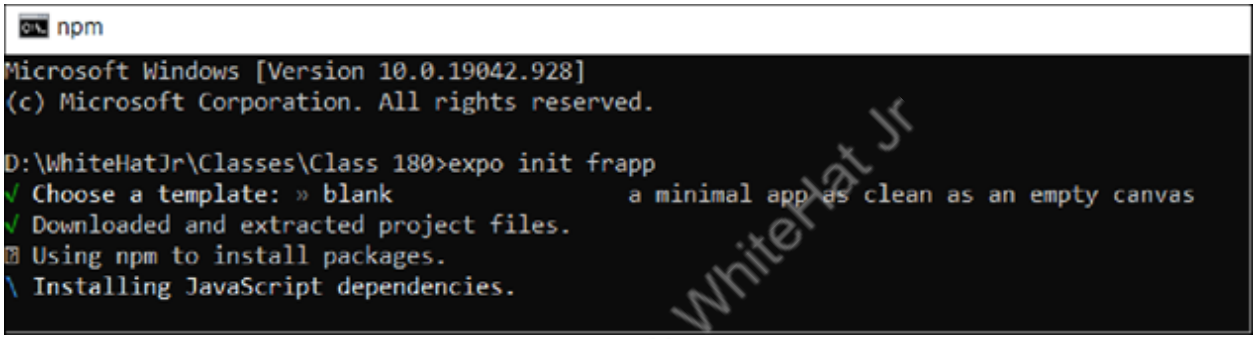
- We learned about face detection

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

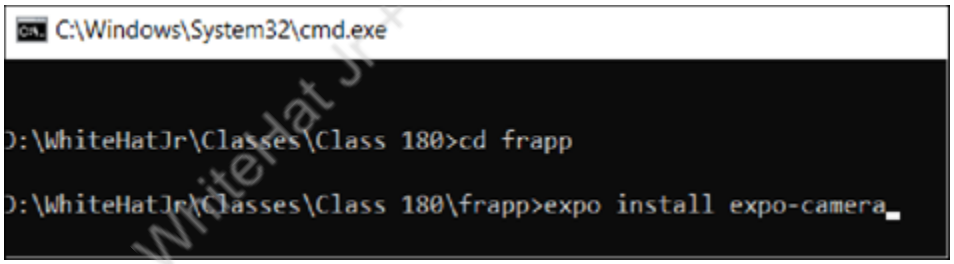
- expo-permissions
- expo-camera
- expo-face-detector
- <Camera/>, <SafeAreaView/> components

## How did we DO the activities?

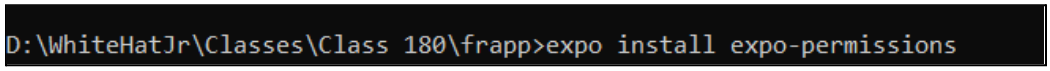
1. Set up the project using expo init



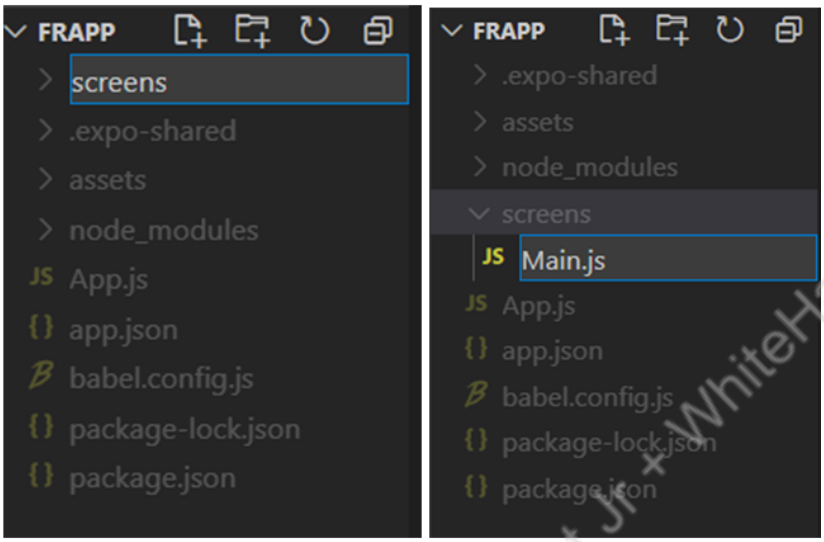

2. Install the following dependencies inside the project folder
   - expo install expo-camera
   - expo install expo-face-detector
   - expo install expo-permissions







3. Create a new folder called **screens** in our root directory and then create a file **Main.js** inside this folder




4. Write the "**Main**" class component and import it in the **App.js**.

```
export default class Main extends React.Component {


}
```

```
import React from 'react';
import Main from "./screens/Main";

export default function App() {
  return (
    <Main />
  )
}
```

5. **Import** react/expo modules in **Main.js**

```
import React from 'react';
import { StyleSheet, Text, View, SafeAreaView, StatusBar, Platform } from 'react-native';
import { Camera } from 'expo-camera';
import * as Permissions from "expo-permissions";
import * as FaceDetector from 'expo-face-detector';
```

6. Set the **state variable** values

```
constructor(props) {
    super(props)
    this.state = {
        hasCameraPermission: null,
        faces: []
    }
}

componentDidMount() {
    Permissions
        .askAsync(Permissions.CAMERA)
        .then(this.onCameraPermission)
}

onCameraPermission = (status) => {
    this.setState({ hasCameraPermission: status.status === 'granted' })
}

onFacesDetected = (faces) => {
    this.setState({ faces: faces })
}

onFaceDetectionError = (error) => {
    console.log(error)
}
```

7. Write the **Main** class component:

- Create StyleSheet for the components to render.

```
const styles = StyleSheet.create({
    container: {
        flex: 1
    },
    droidSafeArea: {
        marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 0
    },
    headingContainer: {
        flex: 0.1,
        alignItems: 'center',
        justifyContent: 'center'
    },
    titleText: {
        fontSize: 30
    },
    cameraStyle: {
        flex: 0.65
    },
    filterContainer: {},
    actionContainer: {}
});
```

- Check if the camera has permissions
- If yes, then render the App Name, Camera and Filters component
- If no, then show a text message with “No access to camera”

```
render() {
    const { hasCameraPermission } = this.state;
    if (hasCameraPermission === null) {
        return <View />
    }
    if (hasCameraPermission === false) {
        return (
            <View style={styles.container}>
                <Text>No access to camera</Text>
            </View>
        )
    }
    console.log(this.state.faces)
    return (
        <View style={styles.container}>
            <SafeAreaView style={styles.droidSafeArea} />
            <View style={styles.headingContainer}>
                <Text style={styles.titleText}>FRAPP</Text>
            </View>
            <View style={styles.cameraStyle}>
                <Camera
                    style={{ flex: 1 }}
                    type={Camera.Constants.Type.front}
                    faceDetectorSettings={{
                        mode: FaceDetector.Constants.Mode.fast,
                        detectLandmarks: FaceDetector.Constants.Landmarks.all,
                        runClassifications: FaceDetector.Constants.Classifications.all
                    }}
                    onFacesDetected={this.onFacesDetected}
                    onFacesDetectionError={this.onFacesDetectionError}
                />
            </View>
            <View style={styles.filterContainer}>
            </View>
            <View style={styles.actionContainer}>
            </View>
        </View>
    )
}
```

8. Test the output: We will get an array of **faces** data.


We have successfully learned to use FaceDetector API in expo.

## What's NEXT?

In the next class, we will learn to apply face filters based on data collected after face detection.

## EXTEND YOUR KNOWLEDGE:

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