```
import React, { Component } from 'react';
import {
    View,
    Text,
    StyleSheet,
    ToastAndroid
} from 'react-native';
import { TextInput } from 'react-native-paper';
import firebase from 'firebase';
import AlgorithmInput from '../../components/AlgorithmInput';
import { TouchableOpacity, ScrollView } from 'react-native-gesture-handler';
import Toast from 'react-native-simple-toast';
import ButtonComponent from '../../components/ButtonComponent';
import { Icon } from 'react-native-elements';


class AdminScreen extends Component {
    constructor(props) {
        super(props);
        this.state = {
            distance: 0,
            nightdrive_multiplier: 0,
            heavyclass: 0,
            middleclass: 0,
            lightclass: 0,
            provisional_licence: 0,
            age_addition: 0,
            age_conditional: 0,
            acceleration_conditional: 0,
            hard_braking_penalty: 0
        }
        this.distance = ''
        this.nightdrive = ''
        this.heavyclass = ''
        this.middleclass = ''
        this.lightclass = ''
        this.provLicence = ''
        this.ageAddition = ''
        this.ageConditional = ''
        this.accelerationConditional = ''
        this.brakingPenalty = ''
      }

    componentDidMount(){
        //fetch firebase algorithm conditions
        firebase.database().ref(`/algorithm/`).once('value').then((snapshot) => {
            this.distance = snapshot.val().distance
            this.nightdrive = snapshot.val().nightdrive_multiplier
            this.heavyclass = snapshot.val().heavyclass
            this.middleclass = snapshot.val().middleclass
            this.lightclass = snapshot.val().lightclass
            this.provLicence = snapshot.val().provisional_licence
            this.ageConditional = snapshot.val().age_conditional
            this.ageAddition = snapshot.val().age_addition
            this.accelerationConditional = snapshot.val().acceleration_conditional
            this.brakingPenalty = snapshot.val().hard_braking_penalty

            this.setState({
                distance: this.distance,
            nightdrive_multiplier: this.nightdrive,
            heavyclass: this.heavyclass,
            middleclass: this.middleclass,
            lightclass: this.lightclass,
            provisional_licence: this.provLicence,
            age_conditional: this.ageConditional,
            age_addition: this.ageAddition,
            acceleration_conditional: this.accelerationConditional,
            hard_braking_penalty: this.brakingPenalty
        })
    })
}

//event listener
submitChanges(){
    const { currentUser } = firebase.auth();
    const {
        distance,
        nightdrive_multiplier,
        heavyclass,
        middleclass,
        lightclass,
        provisional_licence,
        age_conditional,
        age_addition,
        acceleration_conditional,
        hard_braking_penalty
     } = this.state

    var Data = {
        distance: Number(distance),
        nightdrive_multiplier: Number(nightdrive_multiplier),
        heavyclass: Number(heavyclass),
        middleclass: Number(middleclass),
        lightclass: Number(lightclass),
        provisional_licence: Number(provisional_licence),
        age_conditional: Number(age_conditional),
        age_addition: Number(age_addition),
        acceleration_conditional: Number(acceleration_conditional),
        hard_braking_penalty: Number(hard_braking_penalty)
      };

      var updates = {};
      updates[`/algorithm/`] = Data;
      console.log(Data)

      //update firebase /algorithms/ db
      return firebase.database().ref().update(updates)
        .then(result => {
            alert('Changes submitted succesfully...')
        })
        .catch(error => {
            console.log(error.message)
        })
}

render(){
    return(
        <View style={{color: '#003f5c', flex: 1}}>
```

```
                <View style={{flex: 6}}>
                    <View style={{paddingTop: '10%'}}>
                        <Text style={styles.heading}>Adjust Insurance Algorithm</Text>
                    </View>

                    <ScrollView>
                    <View
                        style={{
                            borderBottomColor: 'black',
                            borderBottomWidth: 1,
                        }}
                    />
                    <AlgorithmInput
                        text="Distance"
                        placeholder={this.distance.toString()}
                        onChangeText={(distance) => this.setState({distance})}
                        measurement="cents/km"
                    />

                    <View
                        style={{
                            borderBottomColor: 'black',
                            borderBottomWidth: 1,
                        }}
                    />
                    <AlgorithmInput
                        text="Night Drive"
                        placeholder={this.nightdrive.toString()}
                        onChangeText={(nightdrive_multiplier) =>
this.setState({nightdrive_multiplier})}
                        measurement="% additional"
                    />

                    <View
                        style={{
                            borderBottomColor: 'black',
                            borderBottomWidth: 1,
                        }}
                    />

                    <AlgorithmInput
                        text="Heavy Class"
                        placeholder={this.heavyclass.toString()}
                        onChangeText={(heavyclass) => this.setState({heavyclass})}
                        measurement="% additional"
                    />

                    <AlgorithmInput
                        text="Middle Class"
                        placeholder={this.middleclass.toString()}
                        onChangeText={(middleclass) => this.setState({middleclass})}
                        measurement="% additional"
                    />

                    <AlgorithmInput
                        text="Light Class"
                        placeholder={this.lightclass.toString()}
                        onChangeText={(lightclass) => this.setState({lightclass})}
                        measurement="% additional"
                    />

                    <View
                    style={{
                        borderBottomColor: 'black',
                        borderBottomWidth: 1,
                    }}
                    />

                    <AlgorithmInput
                        text="Provisional Licence"
                        placeholder={this.provLicence.toString()}
                        onChangeText={(provisional_licence) =>
this.setState({provisional_licence})}
                        measurement="% additional"
                    />

                    <View
                    style={{
                        borderBottomColor: 'black',
                        borderBottomWidth: 1,
                    }}
                    />

                    <AlgorithmInput
                        text="If car year is older than"
                        placeholder={this.ageConditional.toString()}
                        onChangeText={(age_conditional) =>
this.setState({age_conditional})}
                        measurement="years then"
                    />

                    <AlgorithmInput
                        text="Add"
                        placeholder={this.ageAddition.toString()}
                        onChangeText={(age_addition) => this.setState({age_addition})}
                        measurement="% additional"
                    />

                    <View
                        style={{
                            borderBottomColor: 'black',
                            borderBottomWidth: 1,
                        }}
                    />

                    <Text style={{width: '80%', paddingTop: 10, alignSelf:
'center'}}>4.2m/s^2 is the recommended aggressive braking criteria</Text>
                    <AlgorithmInput
                        text="Aggressive braking criteria"
                        placeholder={this.accelerationConditional.toString()}
                        onChangeText={(acceleration_conditional) =>
this.setState({acceleration_conditional})}
                        measurement="m/s^2"
                    />

                    <AlgorithmInput
                        text="Aggressive braking penalty"
                        placeholder={this.brakingPenalty.toString()}
                        onChangeText={(hard_braking_penalty) =>
this.setState({hard_braking_penalty})}
                        measurement="cents"
                    />

                    <View
                        style={{
                            borderBottomColor: 'black',
                            borderBottomWidth: 1,
                        }}
                    />
                    </ScrollView>
                </View>

                <View style={{flex: 1, marginBottom: '10%'}}>
                    <ButtonComponent
                        onPress={() => this.submitChanges()}
                        text="Submit Changes"
                        icon="save"
                        type="antdesign"
                    />

                    <ButtonComponent
                        onPress={() => this.props.navigation.goBack()}
                        text="Back"
                        icon="close"
                        type="antdesign"
                    />
                </View>
            </View>
        )
    }
}

const styles = StyleSheet.create({
    heading:{
        textAlign: 'center',
        padding: 20,
        fontSize: 28,
        color: '#2E6CB5'
    }
})

export default AdminScreen;
```