Design a React Native application that includes a feedback form with specific fields and validation, event handling for form submission and clearing.

Describe how to create a feedback form in React Native with the following fields: Name, Email, and Rating.

Explain how to handle the OnSubmit event to validate the form data before submission. Describe the logic for checking that all required fields are filled and that email validation is successful.

Discuss how to handle the OnPress event when the user clicks a "Clear" button to reset the form fields.

Explain how to set a maximum and minimum character limit for the Name field to ensure it falls within an acceptable range.

Explain how to validate the user's rating choice to ensure it falls within a specified range (e.g., 1-5). Describe how to handle situations where a user attempts to submit a rating outside the allowed range.

Step - 1: Run **npx create-expo-app feedbackForm**

Step-2: Run **cd feedbackForm**

Step-3 **npm install react-native-dropdown-picker**

import React, { useState } from 'react';

import { StyleSheet, Text, View, SafeAreaView, TouchableOpacity, TextInput, ImageBackground, Alert } from 'react-native';

import DropDownPicker from 'react-native-dropdown-picker';

const image = { uri: 'https://cdn.pixabay.com/photo/2023/03/21/15/11/ai-generated-7867492\_1280.jpg' };

export default function App() {

const [open, setOpen] = useState(false);

const [value, setValue] = useState(null);

const [name, setName] = useState('');

const [email, setEmail] = useState('');

const handleFormSubmit = () => {

if (!name || !email || !value) {

Alert.alert('Validation Error', 'Please fill in all fields.');

return;

}

if (name.length < 3 || name.length > 15) {

Alert.alert('Validation Error', 'Name must be at least 3 characters long.');

return;

}

if (!isValidEmail(email)) {

Alert.alert('Validation Error', 'Please enter a valid email address.');

return;

}

setName('');

setEmail('');

setValue(null);

// Show a success message or navigate to a success screen

Alert.alert('Success', 'Feedback submitted successfully.');

};

const isValidEmail = (email) => {

// Basic email validation, you can use a more robust regex pattern if needed

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return emailPattern.test(email);

};

const handleClearForm = () => {

setName('');

setEmail('');

setValue(null);

};

return (

<View style={styles.container}>

<ImageBackground source={image} style={styles.image} imageStyle={{ opacity: 0.4 }}>

<Text style={styles.textVal}>Feedback</Text>

<SafeAreaView>

<TextInput

placeholder="Your name here!"

style={styles.input}

value={name}

onChangeText={(text) => setName(text)}

/>

<TextInput

placeholder="Your email here!"

style={styles.input}

value={email}

onChangeText={(text) => setEmail(text)}

keyboardType="email-address"

/>

<DropDownPicker

open={open}

value={value}

items={[

{ label: '1', value: 'excellent' },

{ label: '2', value: 'good' },

{ label: '3', value: 'average' },

{ label: '4', value: 'poor' },

]}

setOpen={setOpen}

setValue={setValue}

containerStyle={{ width: 320, zIndex: 1 }}

style={styles.dropdown}

/>

<View style={styles.to\_style}>

<TouchableOpacity style={styles.btnContainer} onPress={handleFormSubmit}>

<Text style={styles.btn}>Submit</Text>

</TouchableOpacity>

</View>

<View style={styles.to\_style}>

<TouchableOpacity style={styles.btnContainer} onPress={handleClearForm}>

<Text style={styles.btn}>Clear</Text>

</TouchableOpacity>

</View>

</SafeAreaView>

</ImageBackground>

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: '#000',

alignItems: 'center',

justifyContent: 'center',

},

textVal: {

color: '#fff',

textAlign: 'center',

fontSize: 30,

},

image: {

flex: 1,

justifyContent: 'center',

resizeMode: 'cover',

alignSelf: 'stretch',

alignItems: 'center',

},

input: {

height: 60,

width: 320,

marginTop: 10,

marginBottom: 10,

padding: 20,

borderRadius: 30,

backgroundColor: '#fff',

opacity: 0.8,

},

dropdown: {

padding: 20,

borderRadius: 30,

},

btnContainer: {

width: 300,

backgroundColor: '#f8f6fe',

borderRadius: 20,

padding: 10,

opacity: 0.8,

marginTop: 10,

marginBottom: 10,

},

btn: {

fontSize: 18,

color: "#000",

alignSelf: "center",

},

to\_style: {

alignItems: 'center',

}

});