App.js

import React,{useState} from 'react';

import { StatusBar } from 'expo-status-bar';

import {NavigationContainer} from '@react-navigation/native';

import {createStackNavigator} from '@react-navigation/stack';

import {createBottomTabNavigator} from '@react-navigation/bottom-tabs';

import {createMaterialTopTabNavigator} from '@react-navigation/material-top-tabs';

import {Ionicons} from '@expo/vector-icons';

import {createDrawerNavigator,

        DrawerContentScrollView,

        DrawerItemList,

        DrawerItem} from '@react-navigation/drawer';

import HomeScreen from './screens/HomeScreen';

import LoginScreen from './screens/LoginScreen';

import RegisterScreen from './screens/RegisterScreen';

import Practice from './screens/Practice';

import Mocks from './screens/Mocks';

const Drawer = createDrawerNavigator();

function practiceScreen() {

  return(

    <Drawer.Navigator initialRouteName="Practice" >

      <Drawer.Screen name="Practice" component={Practice}/>

      <Drawer.Screen name="Mocks" component={Mocks}/>

    </Drawer.Navigator>

  );

}

const appstack = createBottomTabNavigator();

function AppStack() {

   return(

      <appstack.Navigator initialRouteName="Home" >

        <appstack.Screen name = "Home" component={HomeScreen}  />

        <appstack.Screen name = "PracticeScreen" component={practiceScreen}  />

      </appstack.Navigator>

   );

 }

 const authstack = createStackNavigator();

 function AuthStack(){

   return(

     <authstack.Navigator initialRouteName="Login">

       <authstack.Screen name = "Login" component={LoginScreen} options={{headerShown:false}}/>

       <authstack.Screen name = "Register" component = {RegisterScreen} options={{headerShown:false}}/>

     </authstack.Navigator>

   );

 }

 const Stack = createStackNavigator();

 export default function App(){

   return(

     <NavigationContainer>

       <Stack.Navigator initialRouteName="Auth">

         <Stack.Screen name="Auth" component={AuthStack} options={{headerShown:false}}/>

         <Stack.Screen name="App" component={AppStack} options={{headerShown:false}}/>

       </Stack.Navigator>

     </NavigationContainer>

   );

 }

LoginScreen.js

import React from 'react';

import {View, StyleSheet, Text, Button, TextInput, TouchableOpacity} from 'react-native';

export default class LoginScreen extends React.Component {

    render(){

        return(

            <View style={styles.container}>

                <Text style={{fontSize:20}}>Login to continue{"\n"}</Text>

                <TextInput

                placeholder="Username"

                placeholderTextColor="gray"

                style={styles.input}

                />

                <Text>{"\n"}</Text>

                <TextInput

                placeholder="password"

                placeholderTextColor="gray"

                style={styles.input}

                secureTextEntry={true}

                /><Text>{"\n"} </Text>

                <Button title = "Login" style={{width:100}} onPress={() => this.props.navigation.navigate("App")}/>

                <TouchableOpacity>

        <Text style={{fontSize:20}}>new User?<Text style={{color: 'deepskyblue'}} onPress={()=> this.props.navigation.navigate('Register')}>{" "}Register</Text></Text>

                </TouchableOpacity>

            </View>

        );

    }

}

const styles = StyleSheet.create({

    container: {

        flex:1,

        justifyContent: 'center',

        alignItems: 'center'

    },

    input: {

        backgroundColor: "lightgray",

        width:400,

        padding: 10,

        borderRadius: 15

    }

})

RegisterScreen.js

import React from 'react';

import {View, StyleSheet, Text, Button, TextInput, TouchableOpacity} from 'react-native';

export default class RegisterScreen extends React.Component {

    render(){

        return(

            <View style={styles.container}>

                <Text style = {styles.heading}>Register to start preparing!</Text>

                <TextInput

                placeholder="Full Name"

                placeholderTextColor="gray"

                style={styles.input}

                />

                <Text>{"\n"}</Text>

                <TextInput

                placeholder="Email"

                placeholderTextColor="gray"

                style={styles.input}

                /><Text>{"\n"}</Text>

                <TextInput

                placeholder="Create Password"

                placeholderTextColor="gray"

                style={styles.input}

                secureTextEntry={true}

                /><Text>{"\n"} </Text>

                <Button

                title="Register"

                onPress={()=> this.props.navigation.navigate('App')}

                />

                <TouchableOpacity>

                <Text style={{fontSize:15}}>Already have an account?<Text style={{color: 'deepskyblue'}} onPress={()=> this.props.navigation.navigate('Login')}>{" "}Login</Text></Text>

                </TouchableOpacity>

            </View>

        );

    }

}

const styles = StyleSheet.create({

    container: {

        flex:1,

        justifyContent: 'center',

        alignItems: 'center'

    },

    heading: {

        fontSize: 20,

    },

    input: {

        backgroundColor: "lightgray",

        width:400,

        padding: 10,

        borderRadius: 15

    },

})

Practice.js

import React from 'react';

import { View, StyleSheet, Button, TextInput, ScrollView, Text, TouchableOpacity, FlatList, Modal } from 'react-native';

import { Ionicons, MaterialCommunityIcons } from '@expo/vector-icons';

const qa = [

  {

    id: '1',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 1"

  },

  {

    id: '2',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 3"

  },

  {

    id: '3',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 4"

  },

  {

    id: '4',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 2"

  },

  {

    id: '5',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 3"

  },

]

export default class Practice extends React.Component {

  constructor() {

    super();

    this.state = {

      resultText: "",

      calculationText: "",

      ans: "",

      modalVisible: false,

      totalScore: 0

    };

    this.operations = ['D', '+', '-', '\*', '/'];

  }

  calculateResult() {

    const text = this.state.resultText

    this.setState({

      calculationText: eval(text)

    })

  }

  clearScreen() {

    this.setState({

      resultText: "",

      calculationText: ""

    });

  }

  validate() {

    const text = this.state.resultText

    switch (text.slice(-1)) {

      case '+':

      case '-':

      case '\*':

      case '/':

        return false

    }

    return true

  }

  buttonPressed(text) {

    if (text === '=') {

      return this.validate() && this.calculateResult()

    }

    this.setState({

      resultText: this.state.resultText + text

    })

  }

  operate(operation) {

    switch (operation) {

      case 'D':

        let text = this.state.resultText.split('')

        text.pop()

        this.setState({

          resultText: text.join('')

        })

        break

      case '+':

      case '-':

      case '\*':

      case '/':

        const lastChar = this.state.resultText.split('').pop()

        if (this.operations.indexOf(lastChar) > 0) return

        if (this.state.resultText == "") {

          return

        }

        this.setState({

          resultText: this.state.resultText + operation

        })

    }

  }

  renderQuestions = qa => {

    return (

      <View style={{

        backgroundColor: '#FFF',

        borderRadius: 5,

        padding: 8,

        marginVertical: 8,

      }}>

        <Text>{qa.id+")"} {qa.question} </Text>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option1})}><Text>A. {qa.option1}{"\n"}</Text></TouchableOpacity>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option2})}><Text>B. {qa.option2}{"\n"}</Text></TouchableOpacity>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option3})}><Text>C. {qa.option3}{"\n"}</Text></TouchableOpacity>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option4})}><Text>D. {qa.option4}</Text></TouchableOpacity>

        <Button

          title="check"

          onPress={() => {

            if (this.state.ans === qa.answer) { this.setState({totalScore: this.state.totalScore+1}) }

            else { alert('You have selected the wrong answer, the correct answer is: ' + qa.answer) }

            this.setState({ans: ""})

          }}

        />

      </View>

    );

  }

  render() {

    let rows = [];

    let nums = [[1, 2, 3], [4, 5, 6], [7, 8, 9], ['.', 0, '=']];

    for (let i = 0; i < 4; i++) {

      let row = [];

      for (let j = 0; j < 3; j++) {

        row.push(<TouchableOpacity key={nums[i][j]} style={styles.btn}>

          <Text onPress={() => this.buttonPressed(nums[i][j])} style={styles.btntext}>{nums[i][j]}</Text>

        </TouchableOpacity>);

      }

      rows.push(<View key={i} style={styles.row}>{row}</View>);

    }

    let ops = [];

    for (let i = 0; i < 5; i++) {

      ops.push(<TouchableOpacity key={this.operations[i]} style={styles.btn} onPress={() => this.operate(this.operations[i])}>

        <Text style={[styles.btntext, styles.white]}>{this.operations[i]}</Text>

      </TouchableOpacity>)

    }

    return (

      <View style={styles.container}>

        <View style={styles.header}>

          <TouchableOpacity onPress={() => { this.props.navigation.toggleDrawer() }}><Ionicons name="md-menu" size={24} color="black" style={{ marginLeft: 20 }} /></TouchableOpacity>

          <Text style={{ alignItems: 'center', justifyContent: 'center', fontSize: 15, fontWeight: '500', }}>Practice</Text>

          <Text style={{marginRight: 15}}>Total Score: {this.state.totalScore}/{qa.length}</Text>

        </View>

        <View style = {{flexDirection: 'row', marginLeft: 15,flex: 1}}>

        <View style={styles.calculator}>

            <TouchableOpacity style={{ alignItems: 'center' }} onPress={() => { this.setState({ resultText: "", calculationText: "" }) }}><Text style={{ justifyContent: 'center', marginLeft: 5 }}>Clear</Text></TouchableOpacity>

            <View style={styles.result}>

              <Text style={styles.resultText}>{this.state.resultText}</Text>

            </View>

            <View style={styles.calculation}>

              <Text style={styles.calculationText}>{this.state.calculationText}</Text>

            </View>

            <View style={styles.buttons}>

              <View style={styles.numbers}>

                {rows}

              </View>

              <View style={styles.operations}>

                {ops}

              </View>

            </View>

          </View>

          <View style={{flex: 1}}>

            <FlatList

              data={qa}

              renderItem={({ item }) => this.renderQuestions(item)}

              keyExtractor={item => item.id}

              showsVerticalScrollIndicator={true}

            />

          </View>

        </View>

      </View>

    );

  }

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

  },

  header: {

    paddingTop: 30,

    backgroundColor: '#fff',

    paddingBottom: 10,

    flexDirection: 'row',

    borderBottomWidth: 1,

    borderBottomColor: '#EBECF4',

    shadowColor: '#f0f8ff',

    shadowOffset: { height: 5 },

    shadowOpacity: 0.2,

    shadowRadius: 15,

    zIndex: 10,

    justifyContent: 'space-between'

  },

  calculator: {

    marginRight: 10,

    width: "30%",

    backgroundColor: '#fff',

    height: 250

  },

  row: {

    flexDirection: 'row',

    flex: 1,

    justifyContent: 'space-around',

    alignItems: 'center'

  },

  btntext: {

    fontSize: 30,

  },

  white: {

    color: 'white'

  },

  btn: {

    flex: 1,

    alignItems: 'center',

    alignSelf: 'stretch',

    justifyContent: 'center'

  },

  calculationText: {

    fontSize: 24,

    color: 'white',

  },

  resultText: {

    fontSize: 30,

    color: 'white',

  },

  result: {

    flex: 2,

    backgroundColor: '#8bc2f9',

    justifyContent: 'center',

    alignItems: 'flex-end'

  },

  calculation: {

    flex: 1,

    backgroundColor: '#4ba0f4',

    justifyContent: 'center',

    alignItems: 'flex-end',

    padding: 10

  },

  buttons: {

    flex: 7,

    flexDirection: 'row'

  },

  numbers: {

    flex: 3,

    backgroundColor: '#82868c'

  },

  operations: {

    flex: 1,

    justifyContent: 'space-around',

    alignItems: 'stretch',

    backgroundColor: '#b8babc'

  }

});

Mocks.js

import React from 'react';

import { View, StyleSheet, Button, TextInput, ScrollView, Text, TouchableOpacity, FlatList, Modal } from 'react-native';

import { Ionicons, MaterialCommunityIcons } from '@expo/vector-icons';

const qa = [

  {

    id: '1',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 1"

  },

  {

    id: '2',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 3"

  },

  {

    id: '3',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 4"

  },

  {

    id: '4',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 2"

  },

  {

    id: '5',

    question: "this is a question",

    option1: "option 1",

    option2: "option 2",

    option3: "option 3",

    option4: "option 4",

    answer: "option 3"

  },

]

export default class Mocks extends React.Component {

  constructor() {

    super();

    this.state = {

      resultText: "",

      calculationText: "",

      ans: "",

      modalVisible: false,

      totalScore: 0,

      button : false

    };

    this.operations = ['D', '+', '-', '\*', '/'];

  }

  calculateResult() {

    const text = this.state.resultText

    this.setState({

      calculationText: eval(text)

    })

  }

  clearScreen() {

    this.setState({

      resultText: "",

      calculationText: ""

    });

  }

  validate() {

    const text = this.state.resultText

    switch (text.slice(-1)) {

      case '+':

      case '-':

      case '\*':

      case '/':

        return false

    }

    return true

  }

  buttonPressed(text) {

    if (text === '=') {

      return this.validate() && this.calculateResult()

    }

    this.setState({

      resultText: this.state.resultText + text

    })

  }

  operate(operation) {

    switch (operation) {

      case 'D':

        let text = this.state.resultText.split('')

        text.pop()

        this.setState({

          resultText: text.join('')

        })

        break

      case '+':

      case '-':

      case '\*':

      case '/':

        const lastChar = this.state.resultText.split('').pop()

        if (this.operations.indexOf(lastChar) > 0) return

        if (this.state.resultText == "") {

          return

        }

        this.setState({

          resultText: this.state.resultText + operation

        })

    }

  }

  renderQuestions = qa => {

    return (

      <View style={{

        backgroundColor: '#FFF',

        borderRadius: 5,

        padding: 8,

        marginVertical: 8,

      }}>

        <Text>{qa.id+")"} {qa.question} </Text>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option1})}><Text>A. {qa.option1}{"\n"}</Text></TouchableOpacity>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option2})}><Text>B. {qa.option2}{"\n"}</Text></TouchableOpacity>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option3})}><Text>C. {qa.option3}{"\n"}</Text></TouchableOpacity>

    <TouchableOpacity onPress={()=>this.setState({ans: qa.option4})}><Text>D. {qa.option4}</Text></TouchableOpacity>

        <Button

          title="check"

          onPress={() => {

            if (this.state.ans === qa.answer) { this.setState({totalScore: this.state.totalScore+1});  }

            else { alert('You have selected the wrong answer, the correct answer is: ' + qa.answer);  }

            this.setState({ans: ""})

          }}

        />

      </View>

    );

  }

  render() {

    let rows = [];

    let nums = [[1, 2, 3], [4, 5, 6], [7, 8, 9], ['.', 0, '=']];

    for (let i = 0; i < 4; i++) {

      let row = [];

      for (let j = 0; j < 3; j++) {

        row.push(<TouchableOpacity key={nums[i][j]} style={styles.btn}>

          <Text onPress={() => this.buttonPressed(nums[i][j])} style={styles.btntext}>{nums[i][j]}</Text>

        </TouchableOpacity>);

      }

      rows.push(<View key={i} style={styles.row}>{row}</View>);

    }

    let ops = [];

    for (let i = 0; i < 5; i++) {

      ops.push(<TouchableOpacity key={this.operations[i]} style={styles.btn} onPress={() => this.operate(this.operations[i])}>

        <Text style={[styles.btntext, styles.white]}>{this.operations[i]}</Text>

      </TouchableOpacity>)

    }

    return (

      <View style={styles.container}>

        <View style={styles.header}>

          <TouchableOpacity onPress={() => { this.props.navigation.toggleDrawer() }}><Ionicons name="md-menu" size={24} color="black" style={{ marginLeft: 20 }} /></TouchableOpacity>

          <Text style={{ alignItems: 'center', justifyContent: 'center', fontSize: 15, fontWeight: '500', }}>Mock Test</Text>

          <Text style={{marginRight: 15}}>Total Score: {this.state.totalScore}/{qa.length}</Text>

        </View>

        <View style = {{flexDirection: 'row', marginLeft: 15,flex: 1}}>

        <View style={styles.calculator}>

            <TouchableOpacity style={{ alignItems: 'center' }} onPress={() => { this.setState({ resultText: "", calculationText: "" }) }}><Text style={{ justifyContent: 'center', marginLeft: 5 }}>Clear</Text></TouchableOpacity>

            <View style={styles.result}>

              <Text style={styles.resultText}>{this.state.resultText}</Text>

            </View>

            <View style={styles.calculation}>

              <Text style={styles.calculationText}>{this.state.calculationText}</Text>

            </View>

            <View style={styles.buttons}>

              <View style={styles.numbers}>

                {rows}

              </View>

              <View style={styles.operations}>

                {ops}

              </View>

            </View>

          </View>

          <View style={{flex: 1}}>

            <FlatList

              data={qa}

              renderItem={({ item }) => this.renderQuestions(item)}

              keyExtractor={item => item.id}

              showsVerticalScrollIndicator={true}

            />

          </View>

        </View>

      </View>

    );

  }

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

  },

  header: {

    paddingTop: 30,

    backgroundColor: '#fff',

    paddingBottom: 10,

    flexDirection: 'row',

    borderBottomWidth: 1,

    borderBottomColor: '#EBECF4',

    shadowColor: '#f0f8ff',

    shadowOffset: { height: 5 },

    shadowOpacity: 0.2,

    shadowRadius: 15,

    zIndex: 10,

    justifyContent: 'space-between'

  },

  calculator: {

    marginRight: 10,

    width: "30%",

    backgroundColor: '#fff',

    height: 250

  },

  row: {

    flexDirection: 'row',

    flex: 1,

    justifyContent: 'space-around',

    alignItems: 'center'

  },

  btntext: {

    fontSize: 30,

  },

  white: {

    color: 'white'

  },

  btn: {

    flex: 1,

    alignItems: 'center',

    alignSelf: 'stretch',

    justifyContent: 'center'

  },

  calculationText: {

    fontSize: 24,

    color: 'white',

  },

  resultText: {

    fontSize: 30,

    color: 'white',

  },

  result: {

    flex: 2,

    backgroundColor: '#8bc2f9',

    justifyContent: 'center',

    alignItems: 'flex-end'

  },

  calculation: {

    flex: 1,

    backgroundColor: '#4ba0f4',

    justifyContent: 'center',

    alignItems: 'flex-end',

    padding: 10

  },

  buttons: {

    flex: 7,

    flexDirection: 'row'

  },

  numbers: {

    flex: 3,

    backgroundColor: '#82868c'

  },

  operations: {

    flex: 1,

    justifyContent: 'space-around',

    alignItems: 'stretch',

    backgroundColor: '#b8babc'

  }

});

HomeScreen.js

import React, { useState } from 'react';

import { View, StyleSheet, Text, SectionList, TouchableHighlight, Linking, TouchableOpacity, Button } from 'react-native';

const sections = [

    {

        id: 0,

        title: 'SAT',

        link: 'https://studyabroad.shiksha.com/exams/sat/everything-you-need-to-know-about-sat',

        data: [

            { id: 0, name: 'PrepScholar', link: 'https://www.prepscholar.com/' },

            { id: 1, name: 'Official SAT Website - The College Board', link: 'https://collegereadiness.collegeboard.org/sat' },

            { id: 2, name: 'Khan Academy', link: 'https://www.khanacademy.org/test-prep/sat' },

            { id: 3, name: 'CrackSAT', link: 'https://www.cracksat.net/sat-downloads/' },

        ]

    },

    {

        id: 1,

        title: 'GRE',

        link: 'https://www.ets.org/gre/revised\_general/about',

        data: [

            { id: 4, name: 'Manhattan Prep', link: 'https://www.manhattanprep.com/gre/' },

            { id: 5, name: 'Princeton Review', link: 'https://www.princetonreview.com/grad/gre-test-prep?exid=e0067721-23dc-4d67-8bd1-479eac341e19' },

            { id: 6, name: 'Magoosh', link: 'https://gre.magoosh.com/' },

            { id: 7, name: 'Official guide to the GRE general test', link: 'https://bayanebartar.org/file-dl/library/GRE/The-Official-Guide-to-the-GRE-General-Test.pdf' },

        ]

    },

    {

        id: 2,

        title: 'TOEFL',

        link: 'https://www.toeflgoanywhere.org/what-is-toefl',

        data: [

            { id: 8, name: 'Magoosh', link: 'https://magoosh.com/toefl/2017/best-free-toefl-resources/' },

            { id: 9, name: 'ETS TOEFL free preparation resources', link: 'https://www.toeflgoanywhere.org/free-preparation-resources' },

            { id: 10, name: 'learn4good.com', link: 'https://www.learn4good.com/languages/spec\_english\_toefl.htm' },

        ]

    },

    {

        id: 3,

        title: 'IELTS',

        link: 'https://www.ielts.org/what-is-ielts/ielts-introduction',

        data: [

            { id: 11, name: 'ielts.org', link: 'https://www.ielts.org/' },

            { id: 12, name: 'ielts-simon', link: 'https://ielts-simon.com/' },

            { id: 13, name: 'ielts buddy', link: 'https://www.ieltsbuddy.com/' },

            { id: 14, name: 'ielts Liz Youtube channel', link: 'https://www.youtube.com/user/ieltsliz' },

        ]

    },

    {

        id: 4,

        title: 'GMAT',

        link: 'https://studyabroad.shiksha.com/exams/gmat',

        data: [

            { id: 15, name: 'Manhattan Prep', link: 'https://www.manhattanprep.com/gmat/' },

            { id: 16, name: 'Magoosh', link: 'https://magoosh.com/gmat/practice-tests/free-gmat-practice-test-resources/' },

            { id: 17, name: 'GMAT PDF book', link: 'https://www.pdfdrive.com/gmat-official-guide-2019-e185817808.html' },

        ]

    },

]

export default class HomeScreen extends React.Component {

    renderItem = ({ item }) => {

        return (

            <TouchableHighlight onPress={() => Linking.openURL(item.link)}>

                <Text

                    style={styles.rows}

                > {item.name} </Text>

            </TouchableHighlight>

        )

    }

    renderSectionHeader = ({ section }) => {

        return (

            <TouchableHighlight onPress={() => Linking.openURL(section.link)}>

                <Text style={styles.headers}

                >{section.title}</Text>

            </TouchableHighlight>

        )

    }

    render() {

        return (

            <View style={styles.container}>

               <View style = {{flexDirection:'row',justifyContent:'space-between', width:"100%"}}><Text style={styles.textheading}>International Exams prepration</Text>

                <Button

                title = "Sign Out"

                style = {{position:'absolute',top:10,right:20}}

                onPress={()=>this.props.navigation.navigate('Auth')}

                />

               </View>

                <SectionList

                    style={styles.container}

                    sections={sections}

                    renderItem={this.renderItem}

                    renderSectionHeader={this.renderSectionHeader}

                />

            </View>

        );

    }

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        //justifyContent: 'center',

        //backgroundColor: 'white',

        //alignItems: 'center',

      },

      rows: {

        padding: 10,

        //marginBottom: 5,

        backgroundColor: 'skyblue',

        justifyContent: 'center'

      },

      headers: {

        padding: 15,

        marginBottom: 5,

        backgroundColor: 'blue',

        color: 'white',

        fontWeight: 'bold',

        justifyContent: 'center'

      },

      logincontainer: {

        flexDirection: 'row',

        backgroundColor: 'white',

      },

      buttoncontainer: {

        flexDirection: 'row',

      },

      textheading: {

        fontSize: 35,

        color: 'purple',

        alignItems: 'center',

      }

});