Работа с объектами в React Native

В приложении из практической работы №3: - сделайте из задач объект, состоящий из текста задачи, описания и даты создания; - добавьте кнопку изменения свойств задачи; - добавьте возможность скроллинга приложения.

import React, { useState } from 'react';

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

export default function App() {

  const [inputText, setInputText] = useState('');

  const [inputDescription, setInputDescription] = useState('');

  const [tasks, setTasks] = useState([]);

  const [modalVisible, setModalVisible] = useState(false);

  const [editTaskIndex, setEditTaskIndex] = useState(null);

  const addTask = () => {

    if (inputText.trim() && inputDescription.trim()) {

      const newTask = {

        id: Date.now().toString(),

        text: inputText,

        description: inputDescription,

        date: new Date().toLocaleDateString(),

      };

      setTasks([...tasks, newTask]);

      setInputText('');

      setInputDescription('');

      setModalVisible(false);

    } else {

      Alert.alert("Ошибка", "Введите текст задачи и описание");

    }

  };

  const deleteTask = (id) => {

    setTasks(tasks.filter(task => task.id !== id));

  };

  const openEditModal = (index) => {

    const task = tasks[index];

    setInputText(task.text);

    setInputDescription(task.description);

    setEditTaskIndex(index);

    setModalVisible(true);

  };

  const saveEditedTask = () => {

    if (editTaskIndex !== null) {

      const updatedTasks = tasks.map((task, index) =>

        index === editTaskIndex ? { ...task, text: inputText, description: inputDescription } : task

      );

      setTasks(updatedTasks);

      setInputText('');

      setInputDescription('');

      setEditTaskIndex(null);

      setModalVisible(false);

    }

  };

  return (

    <View style={styles.container}>

      <TouchableOpacity style={styles.button} onPress={() => setModalVisible(true)}>

        <Text style={styles.buttonText}>Добавить задачу</Text>

      </TouchableOpacity>

      <FlatList

        data={tasks}

        renderItem={({ item, index }) => (

          <View style={styles.taskItem}>

            <ScrollView>

              <Text style={styles.taskText}>Задача: {item.text}</Text>

              <Text style={styles.taskText}>Описание: {item.description}</Text>

              <Text style={styles.taskDate}>Дата: {item.date}</Text>

            </ScrollView>

            <TouchableOpacity style={styles.editButton} onPress={() => openEditModal(index)}>

              <Text style={styles.editButtonText}>Изменить</Text>

            </TouchableOpacity>

            <TouchableOpacity style={styles.deleteButton} onPress={() => deleteTask(item.id)}>

              <Text style={styles.deleteButtonText}>Удалить</Text>

            </TouchableOpacity>

          </View>

        )}

        keyExtractor={item => item.id}

      />

      <Modal

        animationType="slide"

        transparent={true}

        visible={modalVisible}

        onRequestClose={() => setModalVisible(false)}

      >

        <View style={styles.modalContainer}>

          <TextInput

            style={styles.input}

            placeholder="Введите текст задачи"

            value={inputText}

            onChangeText={setInputText}

          />

          <TextInput

            style={styles.input}

            placeholder="Введите описание задачи"

            value={inputDescription}

            onChangeText={setInputDescription}

          />

          <View style={styles.buttonContainer}>

            <TouchableOpacity style={styles.button} onPress={editTaskIndex !== null ? saveEditedTask : addTask}>

              <Text style={styles.buttonText}>{editTaskIndex !== null ? 'Сохранить изменения' : 'Добавить задачу'}</Text>

            </TouchableOpacity>

            <TouchableOpacity style={styles.button} onPress={() => setModalVisible(false)}>

              <Text style={styles.buttonText}>Отмена</Text>

            </TouchableOpacity>

          </View>

        </View>

      </Modal>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    justifyContent: 'center',

    padding: 16,

    backgroundColor: '#fff',

  },

  input: {

    height: 40,

    borderColor: '#ccc',

    borderWidth: 1,

    marginBottom: 16,

    width: '80%',

    paddingHorizontal: 8,

  },

  button: {

    backgroundColor: '#007BFF',

    padding: 10,

    borderRadius: 5,

    marginBottom: 16,

    width: '80%',

    alignItems: 'center',

  },

  buttonText: {

    color: '#fff',

    fontSize: 16,

  },

  taskItem: {

    flexDirection: 'row',

    justifyContent: 'space-between',

    alignItems: 'center',

    width: '100%',

    padding: 10,

    borderBottomWidth: 1,

    borderBottomColor: '#ccc',

  },

  taskText: {

    fontSize: 16,

    width: '70%',

  },

  taskDate: {

    fontSize: 12,

    color: 'gray',

  },

  editButton: {

    backgroundColor: '#FFA500',

    padding: 5,

    borderRadius: 5,

  },

  editButtonText: {

    color: '#fff',

    fontSize: 14,

  },

  deleteButton: {

    backgroundColor: '#FF0000',

    padding: 5,

    borderRadius: 5,

  },

  deleteButtonText: {

    color: '#fff',

    fontSize: 14,

  },

  modalContainer: {

    flex: 1,

    justifyContent: 'center',

    alignItems: 'center',

    backgroundColor: 'rgba(0, 0, 0, 0.5)',

  },

  buttonContainer: {

    flexDirection: 'row',

    justifyContent: 'space-between',

    width: '80%',

  },

});

