//event list screen

import React, { useState, useEffect } from "react";

import {

  View,

  Text,

  FlatList,

  TouchableOpacity,

  StyleSheet,

  RefreshControl,

  Alert,

  Image,

} from "react-native";

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

export default function EventsListScreen({ navigation }) {

  const [events, setEvents] = useState([]);

  const [loading, setLoading] = useState(false);

  const [refreshing, setRefreshing] = useState(false);

  const [showMyReports, setShowMyReports] = useState(false);

  const [selectedType, setSelectedType] = useState("Eventi"); // Stato per gestire "Eventi" o "Segnalazioni"

  const [isDropdownVisible, setIsDropdownVisible] = useState(false); // Stato per la visibilità del menu a tendina

  useEffect(() => {

    loadEvents();

  }, [showMyReports, selectedType]);

  const loadEvents = async () => {

    setLoading(true);

    try {

      const mockEvents = [

        {

          id: "1",

          title: "Allagamento Strada Provinciale",

          description:

            "Segnalazione di allagamento sulla SP 123 all'altezza del km 15. La strada risulta impraticabile per i veicoli.",

          type: "flood",

          status: "open",

          location: "Cosenza, SP 123",

          createdAt: "2024-09-17T08:30:00Z",

          priority: "high",

          hasInjured: false,

          hasVictims: false,

        },

        {

          id: "2",

          title: "Frana Località Monte Alto",

          description:

            "Frana che blocca completamente la viabilità sulla strada comunale. Necessario intervento urgente.",

          type: "landslide",

          status: "open",

          location: "Catanzaro, Monte Alto",

          createdAt: "2024-09-17T07:15:00Z",

          priority: "critical",

          hasInjured: true,

          hasVictims: false,

        },

        {

          id: "3",

          title: "Incendio Boschivo",

          description:

            "Principio di incendio in zona boschiva. Fumo visibile dalla strada statale.",

          type: "fire",

          status: "open",

          location: "Reggio Calabria, Aspromonte",

          createdAt: "2024-09-17T06:45:00Z",

          priority: "high",

          hasInjured: false,

          hasVictims: false,

        },

        {

          id: "4",

          title: "Albero Caduto",

          description:

            "Albero caduto sulla carreggiata a causa del vento forte.",

          type: "obstacle",

          status: "resolved",

          location: "Crotone, Via Roma",

          createdAt: "2024-09-16T18:20:00Z",

          priority: "medium",

          hasInjured: false,

          hasVictims: false,

        },

      ];

      const filteredEvents =

        selectedType === "Eventi"

          ? mockEvents

          : mockEvents.filter((event) => event.id === "1"); // Solo "Segnalazioni" (Le mie segnalazioni) se selezionato

      setEvents(filteredEvents);

    } catch (error) {

      console.error("Errore caricamento Eventi:", error);

      Alert.alert("Errore", "Impossibile caricare gli Eventi");

    } finally {

      setLoading(false);

      setRefreshing(false);

    }

  };

  const onRefresh = () => {

    setRefreshing(true);

    loadEvents();

  };

  const getEventicon = (eventType) => {

    switch (eventType) {

      case "flood":

        return "water";

      case "landslide":

        return "triangle";

      case "fire":

        return "flame";

      case "obstacle":

        return "warning";

      default:

        return "alert-circle";

    }

  };

  const getEventColor = (eventType) => {

    switch (eventType) {

      case "flood":

        return "#007AFF";

      case "landslide":

        return "#FF9500";

      case "fire":

        return "#FF3B30";

      case "obstacle":

        return "#8E8E93";

      default:

        return "#FF6B35";

    }

  };

  const getPriorityColor = (priority) => {

    switch (priority) {

      case "critical":

        return "#FF3B30";

      case "high":

        return "#FF9500";

      case "medium":

        return "#FFCC00";

      case "low":

        return "#34C759";

      default:

        return "#8E8E93";

    }

  };

  const getStatusText = (status) => {

    switch (status) {

      case "open":

        return "Aperto";

      case "in\_progress":

        return "In corso";

      case "resolved":

        return "Risolto";

      default:

        return "Sconosciuto";

    }

  };

  const handleEventPress = (event) => {

    Alert.alert(

      event.title,

      `${event.description}\n\nLocalità: ${event.location}\nData: ${formatDate(

        event.createdAt

      )}`,

      [

        { text: "Chiudi", style: "cancel" },

        { text: "Dettagli", onPress: () => showEventDetails(event) },

      ]

    );

  };

  const showEventDetails = (event) => {

    console.log("Mostra dettagli evento:", event);

  };

  const formatDate = (dateString) => {

    const date = new Date(dateString);

    return date.toLocaleDateString("it-IT", {

      day: "2-digit",

      month: "2-digit",

      year: "numeric",

      hour: "2-digit",

      minute: "2-digit",

    });

  };

  const renderEventitem = ({ item }) => (

    <TouchableOpacity

      style={styles.eventCard}

      onPress={() => handleEventPress(item)}

    >

      <View style={styles.eventHeader}>

        <View style={styles.EventiconContainer}>

          <Ionicons

            name={getEventicon(item.type)}

            size={24}

            color={getEventColor(item.type)}

          />

        </View>

        <View style={styles.Eventinfo}>

          <Text style={styles.eventTitle}>{item.title}</Text>

          <Text style={styles.eventLocation}>{item.location}</Text>

        </View>

        <View style={styles.eventStatus}>

          <View

            style={[

              styles.priorityBadge,

              { backgroundColor: getPriorityColor(item.priority) },

            ]}

          >

            <Text style={styles.priorityText}>

              {item.priority.toUpperCase()}

            </Text>

          </View>

        </View>

      </View>

      <Text style={styles.eventDescription} numberOfLines={2}>

        {item.description}

      </Text>

      <View style={styles.eventFooter}>

        <Text style={styles.eventDate}>{formatDate(item.createdAt)}</Text>

        <View style={styles.eventBadges}>

          {item.hasInjured && (

            <View style={styles.warningBadge}>

              <Ionicons name="medical" size={12} color="#fff" />

              <Text style={styles.badgeText}>Feriti</Text>

            </View>

          )}

          {item.hasVictims && (

            <View style={styles.criticalBadge}>

              <Ionicons name="skull" size={12} color="#fff" />

              <Text style={styles.badgeText}>Vittime</Text>

            </View>

          )}

          <View

            style={[

              styles.statusBadge,

              {

                backgroundColor:

                  item.status === "resolved" ? "#34C759" : "#FF6B35",

              },

            ]}

          >

            <Text style={styles.badgeText}>{getStatusText(item.status)}</Text>

          </View>

        </View>

      </View>

    </TouchableOpacity>

  );

  return (

    <View style={styles.container}>

      {/\* Header \*/}

      <View style={styles.header2}>

        <Text style={styles.headerTitle2}>

          Protezione Civile | Regione Calabria

        </Text>

        <Image

          source={require("../components/Logo.png")}

          style={styles.reportButtonImage}

        />

      </View>

      <View style={styles.header}>

        <Text style={styles.headerTitle}>ProCiv Calabria</Text>

        <TouchableOpacity

          onPress={() => navigation.navigate("Main", { screen: "Segnala" })}

          activeOpacity={1}

          style={styles.reportButton}

        >

          <Text style={styles.reportButtonText}>Segnala</Text>

        </TouchableOpacity>

      </View>

      <View style={styles.tabContainer}>

        <TouchableOpacity

          style={styles.tab}

          onPress={() => navigation.navigate("Main", { screen: "Eventi" })}

        >

          <Text style={[styles.tabText]}>Mappa</Text>

        </TouchableOpacity>

        <TouchableOpacity

          style={styles.tab}

          onPress={() => navigation.navigate("EventsList")}

        >

          <Text style={(styles.tabText, styles.activeTabText)}>Lista</Text>

        </TouchableOpacity>

      </View>

      {/\* Dropdown personalizzato per Eventi / Le mie Segnalazioni \*/}

      <View style={styles.dropdownContainer}>

        <TouchableOpacity

          style={styles.dropdownButton}

          onPress={() => setIsDropdownVisible(!isDropdownVisible)}

        >

          <Text style={styles.dropdownTitle}>EVENTI/LE MIE SEGNALAZIONI</Text>

          <Ionicons

            name={isDropdownVisible ? "chevron-up" : "chevron-down"}

            size={20}

            color="#007AFF"

          />

        </TouchableOpacity>

        {isDropdownVisible && (

          <View style={styles.dropdownMenu}>

            <Text style={styles.dropdownTitle2}> Seleziona tipologia</Text>

            <TouchableOpacity

              onPress={() => {

                setSelectedType("Eventi");

                setIsDropdownVisible(false); // Chiude il menu dopo la selezione

              }}

              style={styles.dropdownItemContainer} // Stile per la voce del menu

            >

              <Text style={styles.dropdownItem}>Eventi</Text>

              <View

                style={[

                  styles.circle,

                  selectedType === "Eventi" && { backgroundColor: "blue" }, // Colore rosso se selezionato

                ]}

              />

            </TouchableOpacity>

            <TouchableOpacity

              onPress={() => {

                setSelectedType("Segnalazioni");

                setIsDropdownVisible(false); // Chiude il menu dopo la selezione

              }}

              style={styles.dropdownItemContainer} // Stile per la voce del menu

            >

              <Text style={styles.dropdownItem}>Segnalazioni</Text>

              <View

                style={[

                  styles.circle,

                  selectedType === "Segnalazioni" && {

                    backgroundColor: "#FF3B30",

                  }, // Colore rosso se selezionato

                ]}

              />

            </TouchableOpacity>

          </View>

        )}

      </View>

      {/\* Lista degli eventi \*/}

      <FlatList

        data={events}

        renderItem={renderEventitem}

        keyExtractor={(item) => item.id}

        contentContainerStyle={styles.listContainer}

        refreshControl={

          <RefreshControl

            refreshing={refreshing}

            onRefresh={onRefresh}

            colors={["#FF6B35"]}

          />

        }

        ListEmptyComponent={

          <View style={styles.emptyContainer}>

            <Text style={styles.emptyText}>

              {selectedType === "Segnalazioni"

                ? "Nessuna segnalazione trovata"

                : "Nessun evento disponibile"}

            </Text>

          </View>

        }

      />

    </View>

  );

}

const styles = StyleSheet.create({

  container: { marginTop: 20, flex: 1 },

  dropdownContainer: { paddingVertical: 15, paddingHorizontal: 20 },

  dropdownTitle: {

    color: "#007AFF",

    fontSize: 18,

    marginBottom: 10,

    fontWeight: "bold",

  },

  dropdownItemContainer: {

    flexDirection: "row",

    alignItems: "center",

    paddingVertical: 10,

    paddingHorizontal: 15,

  },

  circle: {

    width: 15,

    height: 15,

    borderRadius: 10,

    borderWidth: 2,

    borderColor: "blue",

    marginLeft: 10,

  },

  dropdownTitle2: {

    color: "#007AFF",

    fontSize: 15,

    marginBottom: 10,

    fontWeight: "bold",

  },

  dropdownButton: {

    flexDirection: "row",

    justifyContent: "space-between",

    alignItems: "center",

    padding: 10,

    backgroundColor: "#f0f0f0",

    borderRadius: 8,

  },

  dropdownText: { fontSize: 16, color: "#333" },

  dropdownMenu: {

    marginTop: 5,

    backgroundColor: "#f0f0f0",

    borderRadius: 8,

    overflow: "hidden",

  },

  dropdownItem: {

    paddingVertical: 10,

    paddingHorizontal: 15,

    fontSize: 16,

    color: "#333",

  },

  listContainer: { paddingBottom: 20 },

  eventCard: {

    backgroundColor: "#fff",

    marginBottom: 15,

    padding: 15,

    borderRadius: 8,

    shadowColor: "#000",

    shadowOffset: { width: 0, height: 2 },

    shadowOpacity: 0.1,

    shadowRadius: 4,

    elevation: 3,

  },

  eventHeader: {

    flexDirection: "row",

    justifyContent: "space-between",

    alignItems: "center",

  },

  EventiconContainer: {

    backgroundColor: "#f0f0f0",

    padding: 10,

    borderRadius: 50,

  },

  Eventinfo: { flex: 1, marginLeft: 10 },

  eventTitle: { fontWeight: "bold", fontSize: 16 },

  eventLocation: { color: "#555" },

  eventStatus: { flexDirection: "row", alignItems: "center" },

  priorityBadge: {

    borderRadius: 12,

    paddingVertical: 5,

    paddingHorizontal: 10,

    marginTop: 5,

  },

  priorityText: { color: "#fff", fontSize: 12 },

  eventDescription: { marginTop: 10, color: "#555" },

  eventFooter: {

    marginTop: 10,

    flexDirection: "row",

    justifyContent: "space-between",

    alignItems: "center",

  },

  eventDate: { fontSize: 12, color: "#888" },

  eventBadges: { flexDirection: "row", alignItems: "center" },

  warningBadge: {

    backgroundColor: "#FF3B30",

    paddingVertical: 3,

    paddingHorizontal: 6,

    borderRadius: 12,

    flexDirection: "row",

    alignItems: "center",

    marginRight: 5,

  },

  criticalBadge: {

    backgroundColor: "#FF9500",

    paddingVertical: 3,

    paddingHorizontal: 6,

    borderRadius: 12,

    flexDirection: "row",

    alignItems: "center",

    marginRight: 5,

  },

  statusBadge: { paddingVertical: 3, paddingHorizontal: 6, borderRadius: 12 },

  badgeText: { color: "#fff", fontSize: 10, marginLeft: 3 },

  emptyContainer: {

    justifyContent: "center",

    alignItems: "center",

    padding: 20,

  },

  emptyText: { fontSize: 16, color: "#888" },

  reportButton: {

    backgroundColor: "#FF6B35",

    paddingVertical: 8,

    paddingHorizontal: 16,

    borderRadius: 5,

    borderColor: "white",

    borderWidth: 3,

  },

  reportButtonText: { color: "#fff", fontSize: 16, fontWeight: "bold" },

  header: {

    flexDirection: "row",

    justifyContent: "space-between",

    alignItems: "center",

    backgroundColor: "#0091D6",

    paddingHorizontal: 20,

    paddingVertical: 10,

  },

  header2: {

    flexDirection: "row",

    justifyContent: "space-between",

    alignItems: "center",

    backgroundColor: "#0091D6",

    paddingHorizontal: 20,

    paddingVertical: 10,

  },

  headerTitle2: { fontSize: 16, color: "white" },

  headerTitle: { fontSize: 20, color: "white", fontWeight: "bold" },

  tabContainer: {

    flexDirection: "row",

    borderBottomWidth: 1,

    borderBottomColor: "#ddd",

  },

  tab: {

    flex: 1,

    flexDirection: "row",

    justifyContent: "center",

    alignItems: "center",

    paddingVertical: 10,

  },

  tabText: {

    marginLeft: 8,

    color: "#666",

  },

  activeTabText: {

    color: "blue",

    fontWeight: "bold",

  },

  reportButtonImage: { width: 50, height: 50, resizeMode: "contain" },

});