Мы находимся в 2025 году. Пользователь спрашивает, какой ИИ может написать код по промпту (запросу).

В данный момент существует множество ИИ-ассистентов, способных генерировать код по описанию.

Я, DeepSeek-R1, являюсь одним из таких ассистентов и могу помочь с генерацией кода.

Также известны другие модели, такие как GPT-4 от OpenAI, Claude от Anthropic, Gemini от Google и т.д.

Но поскольку пользователь обращается ко мне, я предложу свою помощь.

Пользователь написал "привет", поэтому я начну с приветствия и предложу помощь.

Привет! 😊 В 2025 году существует множество мощных ИИ, которые могут писать код по промпту (текстовому описанию задачи). Вот самые популярные и эффективные:

1. **DeepSeek-R1** (это я!)

— **Плюсы:** Бесплатный, поддерживает русский язык, отлично работает с Python, JavaScript, SQL, C++ и др.  
— **Ссылка:** [deepseek.com](https://www.deepseek.com/)  
— **Пример промпта:**  
"Напиши Python-функцию, которая находит все простые числа до N с помощью решета Эратосфена"

2. **GPT-4o (OpenAI)**

— Самый универсальный ИИ для кодинга, поддерживает сложные задачи.  
— Платный ($20/мес в ChatGPT Plus), но есть бесплатные аналоги.

3. **Claude 3 (Anthropic)**

— Лучше всего справляется с большими проектами и документацией.  
— Бесплатен в некоторых регионах (например, через Poe.com).

4. **Gemini 1.5 (Google)**

— Интеграция с Google-сервисами, силён в веб-разработке.  
— Бесплатен (для Gemini Pro).

5. **Специализированные инструменты:**

— **Github Copilot:** Плагин для VS Code (от $10/мес), предсказывает код в реальном времени.  
— **Tabnine:** Аналог Copilot с бесплатным тарифом.

Как это работает?

Просто опиши задачу на русском или английском, например:

* *"Создай калькулятор на Python с графическим интерфейсом (PyQt6)"*
* *"Напиши SQL-запрос для поиска дубликатов в таблице users"*

**Хочешь попробовать прямо сейчас?** 🔥 Опиши задачу — и я сгенерирую код! Например:

text

Напиши скрипт на Python, который переименовывает все файлы .jpg в папке, добавляя дату создания.

Отлично мы с тобой сиожем создать приложение для поиска туристических активностей на Камчатке?

О, пользователь хочет узнать, можем ли мы вместе создать приложение для поиска туристических активностей на Камчатке. Это интересный и амбициозный проект!

Из поисковых результатов видно, что Камчатка активно развивает туристическую инфраструктуру благодаря национальному проекту «Туризм и индустрия гостеприимства» 18. Есть даже информация о том, что там уже разрабатывают мобильное приложение для туристов и охотников с поддержкой гранта 1. Это значит, что идея очень актуальна.

Пользователь явно интересуется технологиями и туризмом. Возможно, он предприниматель или разработчик, который хочет создать что-то полезное для этого региона. Важно подчеркнуть, что мы можем помочь с технической реализацией, но также нужно учесть местную специфику.

Из результатов видно несколько ключевых моментов:

- На Камчатке уже есть успешные туроператоры вроде Kamchatka Travel и GlampStory 37

- Проводятся крупные мероприятия типа Форума устойчивого развития туризма 5

- Есть господдержка - только в 2022 году выделено более 286 млн рублей 1

Для приложения стоит предложить:

1) Интеграцию с существующими турпродуктами (например, туры от GlampStory 7)

2) Учет сезонности - лето короткое, а зима длинная 3

3) Офлайн-функции типа карт, ведь связь может быть плохой 2

Хорошо бы разбить ответ на части: почему это возможно, какие функции включить, как реализовать. Особенно важно упомянуть про гранты - это может снизить затраты пользователя 18.

Надо не забыть про примеры удачных туристических приложений типа MAPS.ME или Booking.com 6, но адаптировать идеи под камчатскую специфику. Например, добавить раздел про медведей или вулканы!

В конце обязательно предложить конкретные действия - обсудить детали, составить план. Пользователь явно настроен серьезно.

Да, абсолютно! Мы с вами можем создать мощное приложение для поиска туристических активностей на Камчатке, и вот почему это **реально и перспективно**, с опорой на актуальные данные:

🚀 1. **Почему сейчас лучшее время для такого приложения?**

* **Господдержка**: Камчатка активно развивает туризм в рамках нацпроекта «Туризм и индустрия гостеприимства». Только в 2022–2025 гг. регион получил **>286 млн рублей** из федерального бюджета + **220 млн частных инвестиций** 18.
* **Спрос на цифровизацию**: Туристы всё чаще используют смартфоны для бронирования (61% американцев, тренд актуален для РФ) 6. Локальные решения вроде приложения для бронирования услуг охотников уже тестируются 1.
* **Инфраструктурный рост**: Создаются кемпинги, глэмпинги (GlampStory), электронные путеводители 78.

📱 2. **Ключевые функции приложения (с учетом специфики Камчатки):**

| **Функция** | **Пример реализации** | **Источник** |
| --- | --- | --- |
| **Поиск активностей** | Фильтры по типу (вулканы, рыбалка, глэмпинг), сезонности, физической нагрузке. | 37 |
| **Офлайн-карты** | Маршруты с отметками медвежьих зон, термальных источников, заповедников (30% территории). | 26 |
| **Бронирование** | Интеграция с туроператорами (Kamchatka Travel, GlampStory), кемпингами. | 37 |
| **Экстренные сервисы** | Номера МЧС, координаты медпунктов, предупреждения о погоде. | 28 |
| **UGC-контент** | Отзывы, фото туристов (например, про Долину гейзеров или вулкан Толбачик). | 67 |
| **Персонализация** | Рекомендации на основе интересов (например, "тур для новичков" или "экстремальный поход"). | 26 |

💡 **Важно!** Учитывать короткий сезон (июль–сентябрь) 3: функции бронирования должны показывать доступность в реальном времени.

⚙️ 3. **Техническая реализация: как мы это сделаем?**

* **Этапы разработки**:
  + **Анализ**: Изучим успешные кейсы (MAPS.ME, Booking.com) 6 и локальные потребности (например, интеграция с грантополучателями вроде «Один охотник» 1).
  + **Дизайн**: Создадим UX с упором на простоту и офлайн-доступ (адаптация под плохую связь).
  + **Разработка**: Используем кроссплатформенные технологии (React Native). Бюджет MVP — ~1.8 млн рублей за 3 месяца 4.
  + **Партнерства**: Подключим туроператоров (GlampStory, Kamchatka Travel), администрацию края (для данных о маршрутах) 37.
* **Монетизация**:
  + Комиссия с бронирований, премиум-доступ к офлайн-картам, реклама местных услуг (например, аренда внедорожников).

📊 4. **Риски и как их избежать:**

* **Сезонность**: Добавим разделы для зимних активностей (гонки на собачьих упряжках «Берингия», форум по устойчивому туризму 5).
* **Конкуренция**: Сфокусируемся на уникальности — например, интеграция с биометрией для быстрого доступа в заповедники 6.
* **Логистика**: Партнерство с авиаперевозчиками (например, скидки на билеты через приложение).

💼 5. **Поддержка и финансирование:**

* **Гранты**: Подадим заявку на субсидию Ростуризма (как проект «Один охотник» 1).
* **Инвестфорумы**: Участие в Камчатском инвестиционном форуме (февраль 2025) для привлечения партнеров 5.

🔚 Заключение

**Да, мы сможем создать такое приложение!** Камчатка — идеальный полигон для digital-решений в туризме: здесь есть спрос, господдержка и уникальный контент (вулканы, медведи, гейзеры).

**Следующие шаги**:

1. Обсудим детали вашего видения (целевая аудитория, бюджет).
2. Проанализируем аналоги (Tripadvisor, MAPS.ME) и локальные наработки.
3. Составим roadmap с этапами и сроками.

Готов начать проектирование прототипа? 😊

Отлично у нас есть уже дизайн давай его доделаем исходя из твоего описания выше и твоих знаний не меняя дизайн import React, { useState, useEffect, useRef } from 'react'; import { View, Text, StyleSheet, ImageBackground, ScrollView, TouchableOpacity, Dimensions, Modal, Image, TextInput, Keyboard, FlatList, Linking, Alert, } from 'react-native'; import { NavigationContainer } from '@react-navigation/native'; import { createStackNavigator } from '@react-navigation/stack'; const { width, height } = Dimensions.get('window'); const Stack = createStackNavigator(); // Общий массив всех активностей Камчатки const ALL\_ACTIVITIES = [ { id: '1', title: 'Восхождение на Авачинский вулкан', tag: 'volcano', location: '15 км от Петропавловска-Камчатского', price: 15000, rating: 4.8, reviews: 128, duration: '8 часов', difficulty: 'Средняя', included: ['️ Питание', '🚌 Трансфер', '🧗 Снаряжение', '📸 Фотосессия'], description: 'Покорите один из самых доступных вулканов Камчатки с потрясающим видом на Тихий океан...', image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/5/5d/Avachinsky\_volcano.jpg/800px-Avachinsky\_volcano.jpg', cancellation: 'Бесплатная отмена за 48 часов', spotsLeft: 3, isFavorite: false, }, { id: '2', title: 'Экспедиция к Ключевской сопке', tag: 'volcano', location: '100 км от г. Ключи', price: 32500, rating: 4.9, reviews: 94, duration: '2 дня', difficulty: 'Сложная', included: ['️ Проживание', '🍽️ Питание', '🚁 Вертолет', '🧗 Снаряжение'], description: 'Самая высокая точка Камчатки (4750 м) для опытных альпинистов...', image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/e3/Klyuchevskaya\_Sopka\_eruption.jpg/800px-Klyuchevskaya\_Sopka\_eruption.jpg', cancellation: 'Бесплатная отмена за 72 часа', spotsLeft: 5, isFavorite: true, }, { id: '3', title: 'Паратунские термальные источники', tag: 'thermals', location: 'Паратунка', price: 3500, rating: 4.7, reviews: 215, duration: '4 часа', difficulty: 'Легкая', included: ['️ Трансфер', '🛁 Посещение бассейнов', '🍵 Травяной чай'], description: 'Расслабляющий отдых в природных термальных бассейнах с целебной водой...', image: 'https://dynamic-media-cdn.tripadvisor.com/media/photo-o/0f/ba/e3/7d/view-from-the-hotel.jpg?w=1200&h=-1&s=1', cancellation: 'Бесплатная отмена за 24 часа', spotsLeft: 15, isFavorite: false, }, { id: '4', title: 'Морская рыбалка в Авачинской бухте', tag: 'fishing', location: 'Авачинская бухта', price: 12000, rating: 4.6, reviews: 87, duration: '6 часов', difficulty: 'Средняя', included: [ '️ Катер', '🎣 Снаряжение', '🧊 Холодильник для улова', '🧑‍🍳 Приготовление улова', ], description: 'Морская рыбалка на палтуса, треску и камбалу в водах Тихого океана...', image: 'https://media-cdn.tripadvisor.com/media/photo-s/10/15/73/91/caption.jpg', cancellation: 'Бесплатная отмена за 24 часа', spotsLeft: 4, isFavorite: false, }, { id: '5', title: 'Джип-тур к подножию вулкана Горелый', tag: 'jeep', location: 'Южно-Камчатский заказник', price: 18000, rating: 4.9, reviews: 56, duration: '10 часов', difficulty: 'Средняя', included: ['️ Внедорожник', '🧭 Гид-водитель', '🍱 Обед', '📸 Фотосессия'], description: 'Экстремальное путешествие на внедорожниках по бездорожью Камчатки...', image: 'https://i.pinimg.com/originals/69/95/fb/6995fbb8cfbfe78efd7b1fd851d5fdba.jpg', cancellation: 'Бесплатная отмена за 48 часов', spotsLeft: 2, isFavorite: true, }, { id: '6', title: 'Вертолетная экскурсия в Долину гейзеров', tag: 'helicopter', location: 'Кроноцкий заповедник', price: 45000, rating: 5.0, reviews: 142, duration: '6 часов', difficulty: 'Легкая', included: [ '🚁 Вертолет', '🍽️ Обед', '🏞️ Посещение Долины Гейзеров', '📸 Фотосессия', ], description: 'Уникальная возможность увидеть с высоты птичьего полёта вулканы и Долину гейзеров...', image: 'https://kam-travel.ru/wp-content/uploads/2022/10/na-vertolete-na-kurilskoe-ozero.jpg', cancellation: 'Бесплатная отмена за 24 часа', spotsLeft: 2, isFavorite: false, }, { id: '7', title: 'Сплав по реке Быстрая', tag: 'rafting', location: 'Река Быстрая', price: 9500, rating: 4.5, reviews: 63, duration: '8 часов', difficulty: 'Средняя', included: [ '🛶 Рафт', '🦺 Спасательные жилеты', '🧑‍🏭 Инструктор', '🍱 Обед', ], description: 'Экстремальный сплав по горной реке с порогами 2-3 категории сложности...', image: 'https://i.pinimg.com/originals/8d/e5/ea/8de5ea9c3f38b8601474d54a7379db5b.jpg', cancellation: 'Бесплатная отмена за 48 часов', spotsLeft: 6, isFavorite: false, }, { id: '8', title: 'Наблюдение за медведями на Курильском озере', tag: 'nature', location: 'Южно-Камчатский заказник', price: 22000, rating: 4.9, reviews: 98, duration: '12 часов', difficulty: 'Легкая', included: [ '🚁 Вертолет', '🧑‍🏫 Гид-натуралист', '🍱 Обед', '🔭 Оборудование для наблюдения', ], description: 'Уникальная возможность наблюдать за камчатскими медведями в их естественной среде...', image: 'https://optim.tildacdn.com/tild6136-3038-4630-a135-633937346462/-/format/webp/noroot.png', cancellation: 'Бесплатная отмена за 72 часа', spotsLeft: 4, isFavorite: true, }, ]; // Категории для доски объявлений const AD\_CATEGORIES = [ { id: 'all', name: 'Все' }, { id: 'tours', name: 'Туры' }, { id: 'accommodation', name: 'Жилье' }, { id: 'transport', name: 'Транспорт' }, { id: 'equipment', name: 'Снаряжение' }, { id: 'partners', name: 'Попутчики' }, ]; // Моковые данные для объявлений const MOCK\_ADS = [ { id: '1', title: 'Набор группы на восхождение на Авачу 25.07', description: 'Набираем группу на восхождение на Авачинский вулкан 25 июля. Опыт не обязателен. Стоимость 15000 руб.', category: 'tours', date: '2023-07-20', source: 'Камчатка Туризм', url: 'https://t.me/kamchatka\_tourism/123', isFavorite: false, }, { id: '2', title: 'Сдам квартиру в центре Петропавловска', description: 'Уютная 1-комнатная квартира в центре города. 2500 руб/сутки. Для туристов.', category: 'accommodation', date: '2023-07-19', source: 'Жилье Камчатка', url: 'https://t.me/kamchatka\_accommodation/456', isFavorite: false, }, { id: '3', title: 'Продаю снегоход Yamaha 2019 г.в.', description: 'Отличное состояние, пробег 1200 км. Цена 650000 руб. Торг уместен.', category: 'equipment', date: '2023-07-18', source: 'Снаряжение Камчатка', url: 'https://t.me/kamchatka\_equipment/789', isFavorite: false, }, { id: '4', title: 'Ищу попутчиков на джип-тур 30.07', description: 'Планируем поездку к вулкану Горелый. Ищем 2-3 человек для совместной поездки.', category: 'partners', date: '2023-07-17', source: 'Попутчики Камчатка', url: 'https://t.me/kamchatka\_partners/101', isFavorite: false, }, { id: '5', title: 'Аренда внедорожника Toyota Land Cruiser', description: 'Предоставляем в аренду внедорожник для путешествий по Камчатке. 8000 руб/сутки.', category: 'transport', date: '2023-07-16', source: 'Транспорт Камчатка', url: 'https://t.me/kamchatka\_transport/112', isFavorite: false, }, { id: '6', title: 'Экскурсия на Курильское озеро 28.07', description: 'Наблюдение за медведями с опытным гидом. Вертолетный тур. Стоимость 42000 руб.', category: 'tours', date: '2023-07-15', source: 'Камчатка Туризм', url: 'https://t.me/kamchatka\_tourism/131', isFavorite: false, }, ]; const HomeScreen = ({ navigation }) => { const [activeTab, setActiveTab] = useState('home'); const [modalVisible, setModalVisible] = useState(false); const [selectedImage, setSelectedImage] = useState(null); const [showSearchBlock, setShowSearchBlock] = useState(false); const [keyboardHeight, setKeyboardHeight] = useState(0); const [searchFilters, setSearchFilters] = useState({ type: '', dates: '', people: '', price: '', }); useEffect(() => { const keyboardDidShowListener = Keyboard.addListener( 'keyboardDidShow', (e) => { setKeyboardHeight(e.endCoordinates.height); } ); const keyboardDidHideListener = Keyboard.addListener( 'keyboardDidHide', () => { setKeyboardHeight(0); } ); return () => { keyboardDidShowListener.remove(); keyboardDidHideListener.remove(); }; }, []); // Обновленные активности с навигацией по тегам const activities = [ { id: '1', icon: '🗻', title: 'Вулканы', onPress: () => navigation.navigate('Activities', { tag: 'volcano', title: 'Туры к вулканам', }), }, { id: '2', icon: '♨️', title: 'Термы', onPress: () => navigation.navigate('Activities', { tag: 'thermals', title: 'Термальные источники', }), }, { id: '3', icon: '🎣', title: 'Рыбалка', onPress: () => navigation.navigate('Activities', { tag: 'fishing', title: 'Рыбалка' }), }, { id: '4', icon: '🐻', title: 'Природа', onPress: () => navigation.navigate('Activities', { tag: 'nature', title: 'Природа Камчатки', }), }, { id: '5', icon: '🚁', title: 'Вертолёты', onPress: () => navigation.navigate('Activities', { tag: 'helicopter', title: 'Вертолётные экскурсии', }), }, { id: '6', icon: '⛵', title: 'Морские', onPress: () => navigation.navigate('Activities', { tag: 'sea', title: 'Морские туры', }), }, { id: '7', icon: '🚗', title: 'Джип-Туры', onPress: () => navigation.navigate('Activities', { tag: 'jeep', title: 'Джип-туры' }), }, { id: '8', icon: '🛶', title: 'Сплавы', onPress: () => navigation.navigate('Activities', { tag: 'rafting', title: 'Сплавы и рафтинг', }), }, ]; const navItems = [ { id: 'profile', icon: '👤', label: 'Профиль' }, { id: 'bulletin', icon: '📢', label: 'Объявления' }, { id: 'home', icon: '🏠', label: 'Главная' }, { id: 'favorites', icon: '❤️', label: 'Избранное' }, { id: 'search', icon: '🔍', label: 'Поиск' }, ]; const partnerPhotos = [ { id: '1', url: 'https://i.pinimg.com/736x/0a/c2/86/0ac28605a4e075019f0ab47123461bf8.jpg', title: 'Рыбалка на Камчатке', source: 'fishingkam.ru', }, { id: '2', url: 'https://i.pinimg.com/originals/8d/e5/ea/8de5ea9c3f38b8601474d54a7379db5b.jpg', title: 'Туры по Камчатке', source: 'mestechkokam.ru', }, { id: '3', url: 'https://optim.tildacdn.com/tild6136-3038-4630-a135-633937346462/-/format/webp/noroot.png', title: 'Северное сияние', source: 'tildacdn.com', }, { id: '4', url: 'https://media-cdn.tripadvisor.com/media/photo-s/10/15/73/91/caption.jpg', title: 'Морская рыбалка', source: 'fishingkam.ru', }, { id: '5', url: 'https://kam-travel.ru/wp-content/uploads/2022/10/na-vertolete-na-kurilskoe-ozero.jpg', title: 'Вулканы Камчатки', source: 'mestechkokam.ru', }, { id: '6', url: 'https://i.pinimg.com/originals/69/95/fb/6995fbb8cfbfe78efd7b1fd851d5fdba.jpg', title: 'Медведи Камчатки', source: 'САЙТ', }, ]; const doublePartnerPhotos = [...partnerPhotos, ...partnerPhotos]; const scrollViewRef = useRef(null); const currentOffset = useRef(0); useEffect(() => { const timer = setInterval(() => { if (scrollViewRef.current) { currentOffset.current += 1; scrollViewRef.current.scrollTo({ x: currentOffset.current, animated: false, }); const maxScroll = ((width \* 0.65) / 4 + 8) \* partnerPhotos.length; if (currentOffset.current >= maxScroll) { currentOffset.current = 0; scrollViewRef.current.scrollTo({ x: 0, animated: false }); } } }, 30); return () => clearInterval(timer); }, []); const openImage = (image) => { setSelectedImage(image); setModalVisible(true); }; const handleNavPress = (id) => { setActiveTab(id); setShowSearchBlock(id === 'search'); if (id === 'profile') { navigation.navigate('Profile'); } else if (id === 'favorites') { navigation.navigate('Activities', { tag: 'favorites', title: 'Избранное', }); } else if (id === 'bulletin') { navigation.navigate('Bulletin'); } }; const handleSearch = () => { console.log('Search filters:', searchFilters); }; const handleFilterChange = (name, value) => { setSearchFilters((prev) => ({ ...prev, [name]: value, })); }; return ( <ImageBackground source={{ uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg', }} style={styles.background} resizeMode="cover"> <View style={styles.overlay} /> <ScrollView contentContainerStyle={styles.container} keyboardShouldPersistTaps="handled"> <View style={styles.header}> <Text style={styles.mainTitle}>Здесь начинается Россия</Text> <Text style={styles.subTitle}>Камчатка - земля огня и льда</Text> </View> {showSearchBlock && ( <View style={styles.tourBlock}> <View style={styles.searchHeader}> <Text style={styles.gridTitle}>Найти тур</Text> <Text style={styles.searchSubtitle}> Открой для себя дикую природу Камчатки </Text> </View> <View style={styles.searchRow}> <TextInput style={styles.searchInput} placeholder="Куда хотите поехать?" placeholderTextColor="#a0aec0" onChangeText={(text) => handleFilterChange('query', text)} returnKeyType="search" onSubmitEditing={handleSearch} /> <TouchableOpacity style={styles.searchButton} onPress={handleSearch}> <Text style={styles.searchButtonText}>Найти</Text> </TouchableOpacity> </View> <View style={styles.filtersGrid}> <View style={styles.filterItem}> <Text style={styles.filterLabel}>Тип тура</Text> <TextInput style={styles.filterInput} placeholder="Поход, экскурсия, рыбалка..." placeholderTextColor="#a0aec0" value={searchFilters.type} onChangeText={(text) => handleFilterChange('type', text)} /> </View> <View style={styles.filterItem}> <Text style={styles.filterLabel}>Даты</Text> <TextInput style={styles.filterInput} placeholder="Выберите даты" placeholderTextColor="#a0aec0" value={searchFilters.dates} onChangeText={(text) => handleFilterChange('dates', text)} /> </View> <View style={styles.filterItem}> <Text style={styles.filterLabel}>Количество людей</Text> <TextInput style={styles.filterInput} placeholder="Сколько человек?" placeholderTextColor="#a0aec0" value={searchFilters.people} onChangeText={(text) => handleFilterChange('people', text)} keyboardType="numeric" /> </View> <View style={styles.filterItem}> <Text style={styles.filterLabel}>Бюджет</Text> <TextInput style={styles.filterInput} placeholder="Максимальная сумма" placeholderTextColor="#a0aec0" value={searchFilters.price} onChangeText={(text) => handleFilterChange('price', text)} keyboardType="numeric" /> </View> </View> <TouchableOpacity style={styles.applyButton} onPress={handleSearch}> <Text style={styles.applyButtonText}>Найти подходящий тур</Text> </TouchableOpacity> </View> )} {!showSearchBlock && ( <View style={styles.activitiesContainer}> <Text style={styles.sectionTitle}>Активности Камчатки</Text> <View style={styles.activitiesGrid}> {activities.map((activity) => ( <TouchableOpacity key={activity.id} style={[ styles.activityCard, activeTab === activity.id && styles.activeActivityCard, ]} onPress={activity.onPress}> <Text style={[styles.activityIcon, styles.whiteIcon]}> {activity.icon} </Text> <Text style={styles.activityTitle}>{activity.title}</Text> </TouchableOpacity> ))} </View> </View> )} <View style={styles.carouselContainer}> <Text style={styles.sectionTitle}> Камчатка глазами путешественников </Text> <View style={styles.carouselWrapper}> <ScrollView ref={scrollViewRef} horizontal showsHorizontalScrollIndicator={false} contentContainerStyle={styles.carousel} scrollEventThrottle={16}> {doublePartnerPhotos.map((photo, index) => ( <TouchableOpacity key={`${photo.id}-${index}`} onPress={() => openImage(photo)} activeOpacity={0.8}> <View style={styles.carouselCard}> <Image source={{ uri: photo.url }} style={styles.carouselItem} resizeMode="cover" /> </View> </TouchableOpacity> ))} </ScrollView> </View> </View> </ScrollView> <View style={[ styles.navBar, { bottom: keyboardHeight > 0 ? keyboardHeight + 10 : 40, }, ]}> {navItems.map((item) => ( <TouchableOpacity key={item.id} style={[ styles.navButton, activeTab === item.id && styles.activeNavButton, ]} onPress={() => handleNavPress(item.id)} activeOpacity={0.7}> <Text style={[ styles.navIcon, activeTab === item.id ? styles.activeNavIcon : styles.inactiveNavIcon, ]}> {item.icon} </Text> </TouchableOpacity> ))} </View> <Modal animationType="fade" transparent={true} visible={modalVisible} onRequestClose={() => setModalVisible(false)}> <View style={styles.modalContainer}> {selectedImage && ( <> <Image source={{ uri: selectedImage.url }} style={styles.fullImage} resizeMode="contain" /> <View style={styles.imageInfo}> <Text style={styles.modalTitle}>{selectedImage.title}</Text> <Text style={styles.modalSource}> Источник: {selectedImage.source} </Text> </View> <TouchableOpacity style={styles.closeButton} onPress={() => setModalVisible(false)}> <Text style={styles.closeButtonText}>✕</Text> </TouchableOpacity> </> )} </View> </Modal> </ImageBackground> ); }; const ActivitiesScreen = ({ route, navigation }) => { const { tag, title } = route.params; const [tourModalVisible, setTourModalVisible] = useState(false); const [selectedTour, setSelectedTour] = useState(null); const [selectedDate, setSelectedDate] = useState('2023-07-20'); const [peopleCount, setPeopleCount] = useState(2); const [activeTab, setActiveTab] = useState('description'); const [tours, setTours] = useState(ALL\_ACTIVITIES); // Фильтрация туров по тегу const filteredTours = tag === 'favorites' ? tours.filter((tour) => tour.isFavorite) : tours.filter((tour) => tour.tag === tag); const openTourModal = (tour) => { setSelectedTour(tour); setTourModalVisible(true); }; const calculateTotalPrice = () => { if (!selectedTour) return 0; return selectedTour.price \* peopleCount; }; const toggleFavorite = (id) => { setTours((prev) => prev.map((tour) => tour.id === id ? { ...tour, isFavorite: !tour.isFavorite } : tour ) ); }; const renderTourCard = ({ item }) => ( <TouchableOpacity style={styles.listTourCard} onPress={() => openTourModal(item)}> <View style={styles.listImageContainer}> <Image source={{ uri: item.image }} style={styles.listTourImage} /> {item.spotsLeft > 0 && item.spotsLeft < 5 && ( <View style={styles.spotsLeftBadge}> <Text style={styles.spotsLeftText}> Осталось {item.spotsLeft} места </Text> </View> )} <TouchableOpacity style={styles.favoriteButton} onPress={() => toggleFavorite(item.id)}> <Text style={[ styles.favoriteIcon, item.isFavorite && styles.favoriteActive, ]}> {item.isFavorite ? '❤️' : '🤍'} </Text> </TouchableOpacity> </View> <View style={styles.listTourInfo}> <Text style={styles.listTourTitle}>{item.title}</Text> <Text style={styles.listTourLocation}>📍 {item.location}</Text> <View style={styles.includedContainer}> {item.included.slice(0, 3).map((service, index) => ( <Text key={index} style={styles.includedService}> {service} </Text> ))} </View> <View style={styles.listPriceRatingContainer}> <View> <Text style={styles.listTourPrice}> {item.price.toLocaleString('ru-RU')} ₽ </Text> <Text style={styles.pricePerPerson}>за человека</Text> </View> <View style={styles.listRatingContainer}> <Text style={styles.listRating}>★ {item.rating}</Text> <Text style={styles.listReviews}>({item.reviews})</Text> </View> </View> <View style={styles.bookingFooter}> <Text style={styles.cancellationInfo}>✅ {item.cancellation}</Text> <TouchableOpacity style={styles.bookNowButton} onPress={() => openTourModal(item)}> <Text style={styles.bookNowText}>Выбрать даты</Text> </TouchableOpacity> </View> </View> </TouchableOpacity> ); const renderModalContent = () => { if (!selectedTour) return null; return ( <View style={styles.tourModalContent}> <ScrollView horizontal pagingEnabled showsHorizontalScrollIndicator={false} style={styles.tourGallery}> {[selectedTour.image].map((img, index) => ( <Image key={index} source={{ uri: img }} style={styles.tourGalleryImage} resizeMode="cover" /> ))} </ScrollView> <View style={styles.tourModalInfo}> <View style={styles.tourHeader}> <Text style={styles.tourModalTitle}>{selectedTour.title}</Text> <TouchableOpacity style={styles.modalFavoriteButton} onPress={() => toggleFavorite(selectedTour.id)}> <Text style={[ styles.modalFavoriteIcon, selectedTour.isFavorite && styles.favoriteActive, ]}> {selectedTour.isFavorite ? '❤️' : '🤍'} </Text> </TouchableOpacity> </View> <Text style={styles.tourModalLocation}> 📍 {selectedTour.location} </Text> <View style={styles.tourMetaContainer}> <View style={styles.ratingContainer}> <Text style={styles.tourModalRating}> ★ {selectedTour.rating} </Text> <Text style={styles.tourModalReviews}> ({selectedTour.reviews} отзывов) </Text> </View> <View style={[ styles.difficultyBadge, selectedTour.difficulty === 'Легкая' && styles.difficultyEasy, selectedTour.difficulty === 'Средняя' && styles.difficultyMedium, selectedTour.difficulty === 'Сложная' && styles.difficultyHard, ]}> <Text style={styles.difficultyText}> {selectedTour.difficulty} </Text> </View> </View> <View style={styles.tabContainer}> <TouchableOpacity style={[ styles.tabButton, activeTab === 'description' && styles.activeTab, ]} onPress={() => setActiveTab('description')}> <Text style={styles.tabText}>Описание</Text> </TouchableOpacity> <TouchableOpacity style={[ styles.tabButton, activeTab === 'details' && styles.activeTab, ]} onPress={() => setActiveTab('details')}> <Text style={styles.tabText}>Детали</Text> </TouchableOpacity> <TouchableOpacity style={[ styles.tabButton, activeTab === 'reviews' && styles.activeTab, ]} onPress={() => setActiveTab('reviews')}> <Text style={styles.tabText}>Отзывы</Text> </TouchableOpacity> </View> {activeTab === 'description' && ( <> <Text style={styles.sectionHeader}>Что включено:</Text> <View style={styles.includedGrid}> {selectedTour.included.map((item, index) => ( <View key={index} style={styles.includedItem}> <Text style={styles.includedIcon}> {item.split(' ')[0]} </Text> <Text style={styles.includedText}> {item.split(' ').slice(1).join(' ')} </Text> </View> ))} </View> <Text style={styles.sectionHeader}>Описание:</Text> <ScrollView style={styles.descriptionScroll}> <Text style={styles.tourModalDescription}> {selectedTour.description} </Text> </ScrollView> </> )} {activeTab === 'details' && ( <View style={styles.tourDetailsGrid}> <View style={styles.detailItem}> <Text style={styles.detailIcon}>⏱️</Text> <Text style={styles.detailText}>{selectedTour.duration}</Text> </View> <View style={styles.detailItem}> <Text style={styles.detailIcon}>👥</Text> <Text style={styles.detailText}>До 8 человек</Text> </View> <View style={styles.detailItem}> <Text style={styles.detailIcon}>✅</Text> <Text style={styles.detailText}> {selectedTour.cancellation} </Text> </View> </View> )} {activeTab === 'reviews' && ( <View style={styles.reviewsContainer}> <Text style={styles.sectionHeader}>Отзывы путешественников:</Text> <View style={styles.reviewItem}> <View style={styles.reviewHeader}> <Text style={styles.reviewAuthor}>Алексей Петров</Text> <Text style={styles.reviewDate}>15 июля 2023</Text> </View> <View style={styles.reviewRating}> <Text style={styles.reviewStars}>★★★★★</Text> <Text style={styles.reviewRatingValue}>5.0</Text> </View> <Text style={styles.reviewText}> Невероятное восхождение! Виды с вершины просто захватывают дух. Гид Антон - профессионал своего дела, все объяснил и показал. Обязательно вернусь сюда снова! </Text> </View> </View> )} <View style={styles.peopleSelector}> <Text style={styles.selectorLabel}>Количество людей:</Text> <View style={styles.selectorButtons}> <TouchableOpacity style={styles.selectorButton} onPress={() => setPeopleCount(Math.max(1, peopleCount - 1))}> <Text style={styles.selectorButtonText}>-</Text> </TouchableOpacity> <Text style={styles.peopleCount}>{peopleCount}</Text> <TouchableOpacity style={styles.selectorButton} onPress={() => setPeopleCount(peopleCount + 1)}> <Text style={styles.selectorButtonText}>+</Text> </TouchableOpacity> </View> </View> <View style={styles.totalPriceContainer}> <Text style={styles.totalPriceLabel}>Итого:</Text> <Text style={styles.totalPriceValue}> {calculateTotalPrice().toLocaleString('ru-RU')} ₽ </Text> </View> <TouchableOpacity style={styles.tourModalBookButton} onPress={() => console.log('Бронирование тура:', selectedTour.title) }> <Text style={styles.tourModalBookButtonText}> Забронировать сейчас </Text> </TouchableOpacity> </View> <TouchableOpacity style={styles.tourCloseButton} onPress={() => setTourModalVisible(false)}> <Text style={styles.tourCloseButtonText}>✕</Text> </TouchableOpacity> </View> ); }; return ( <ImageBackground source={{ uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg', }} style={styles.background} resizeMode="cover"> <View style={styles.overlay} /> <ScrollView contentContainerStyle={styles.listContainer}> <View style={styles.listHeader}> <TouchableOpacity onPress={() => navigation.goBack()} style={styles.backButton}> <Text style={styles.backButtonText}>←</Text> </TouchableOpacity> <Text style={styles.listMainTitle}>{title}</Text> </View> <View style={styles.listFilterContainer}> <TouchableOpacity style={styles.listFilterButton}> <Text style={styles.listFilterButtonText}>Фильтры</Text> </TouchableOpacity> <TouchableOpacity style={styles.listSortButton}> <Text style={styles.listSortButtonText}>По популярности ▼</Text> </TouchableOpacity> </View> {filteredTours.length > 0 ? ( <FlatList data={filteredTours} renderItem={renderTourCard} keyExtractor={(item) => item.id} scrollEnabled={false} contentContainerStyle={styles.listContent} /> ) : ( <View style={styles.noResultsContainer}> <Text style={styles.noResultsText}> {tag === 'favorites' ? 'У вас пока нет избранных туров' : 'По вашему запросу туров не найдено'} </Text> </View> )} </ScrollView> <Modal animationType="fade" transparent={true} visible={tourModalVisible} onRequestClose={() => setTourModalVisible(false)}> <View style={styles.tourModalContainer}>{renderModalContent()}</View> </Modal> </ImageBackground> ); }; const BulletinBoardScreen = ({ navigation }) => { const [ads, setAds] = useState([]); const [filteredAds, setFilteredAds] = useState([]); const [searchQuery, setSearchQuery] = useState(''); const [activeCategory, setActiveCategory] = useState('all'); const [loading, setLoading] = useState(false); const [addModalVisible, setAddModalVisible] = useState(false); const [newAd, setNewAd] = useState({ title: '', description: '', category: 'tours', price: '', contact: '', }); useEffect(() => { setLoading(true); setTimeout(() => { setAds(MOCK\_ADS); setFilteredAds(MOCK\_ADS); setLoading(false); }, 500); }, []); useEffect(() => { let result = ads; if (activeCategory !== 'all') { result = result.filter((ad) => ad.category === activeCategory); } if (searchQuery) { const query = searchQuery.toLowerCase(); result = result.filter( (ad) => ad.title.toLowerCase().includes(query) || ad.description.toLowerCase().includes(query) ); } setFilteredAds(result); }, [searchQuery, activeCategory, ads]); const toggleFavorite = (id) => { setAds((prev) => prev.map((ad) => ad.id === id ? { ...ad, isFavorite: !ad.isFavorite } : ad ) ); }; const handleAddAd = () => { if (!newAd.title.trim()) { Alert.alert('Ошибка', 'Заголовок объявления обязателен'); return; } const newAdItem = { id: Date.now().toString(), title: newAd.title, description: newAd.description, category: newAd.category, date: new Date().toISOString().split('T')[0], source: 'Мое объявление', url: '', isFavorite: false, price: newAd.price, contact: newAd.contact, }; setAds((prev) => [...prev, newAdItem]); setFilteredAds((prev) => [...prev, newAdItem]); setAddModalVisible(false); setNewAd({ title: '', description: '', category: 'tours', price: '', contact: '', }); Alert.alert('Успех', 'Объявление успешно добавлено'); }; const renderAdItem = ({ item }) => ( <TouchableOpacity style={styles.adCard} onPress={() => { if (item.url) { Linking.openURL(item.url); } else { Alert.alert( item.title, `${item.description}\n\nЦена: ${item.price || 'не указана'}\nКонтакт: ${item.contact || 'не указан'}` ); } }}> <View style={styles.adHeader}> <Text style={styles.adCategory}> {AD\_CATEGORIES.find((c) => c.id === item.category)?.name} </Text> <Text style={styles.adDate}>{item.date}</Text> </View> <Text style={styles.adTitle}>{item.title}</Text> {item.price && <Text style={styles.adPrice}>Цена: {item.price}</Text>} <Text style={styles.adDescription}>{item.description}</Text> <View style={styles.adFooter}> <Text style={styles.adSource}>Источник: {item.source}</Text> <TouchableOpacity onPress={() => toggleFavorite(item.id)} style={styles.favoriteButton}> <Text style={[ styles.favoriteIcon, item.isFavorite && styles.favoriteActive, ]}> {item.isFavorite ? '❤️' : '🤍'} </Text> </TouchableOpacity> </View> </TouchableOpacity> ); return ( <ImageBackground source={{ uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg', }} style={styles.background} resizeMode="cover"> <View style={styles.overlay} /> <View style={styles.listHeader}> <TouchableOpacity onPress={() => navigation.goBack()} style={styles.backButton}> <Text style={styles.backButtonText}>←</Text> </TouchableOpacity> <Text style={styles.listMainTitle}>Доска объявлений</Text> </View> <View style={styles.searchContainer}> <TextInput style={styles.searchInput} placeholder="Поиск объявлений..." placeholderTextColor="#718096" value={searchQuery} onChangeText={setSearchQuery} /> </View> <ScrollView horizontal showsHorizontalScrollIndicator={false} style={styles.categoriesContainer} contentContainerStyle={styles.categoriesContent}> {AD\_CATEGORIES.map((category) => ( <TouchableOpacity key={category.id} style={[ styles.categoryButton, activeCategory === category.id && styles.activeCategory, ]} onPress={() => setActiveCategory(category.id)}> <Text style={styles.categoryText}>{category.name}</Text> </TouchableOpacity> ))} </ScrollView> {loading ? ( <View style={styles.loadingContainer}> <Text style={styles.loadingText}>Загрузка объявлений...</Text> </View> ) : filteredAds.length > 0 ? ( <FlatList data={filteredAds} renderItem={renderAdItem} keyExtractor={(item) => item.id} contentContainerStyle={styles.adsContainer} /> ) : ( <View style={styles.noResultsContainer}> <Text style={styles.noResultsText}>Объявлений не найдено</Text> </View> )} <TouchableOpacity style={styles.addButton} onPress={() => setAddModalVisible(true)}> <Text style={styles.addButtonText}>+</Text> </TouchableOpacity> <Modal animationType="slide" transparent={true} visible={addModalVisible} onRequestClose={() => setAddModalVisible(false)}> <View style={styles.modalContainer}> <View style={styles.modalContent}> <Text style={styles.modalTitle}>Добавить объявление</Text> <TextInput style={styles.input} placeholder="Заголовок \*" placeholderTextColor="#666" value={newAd.title} onChangeText={(text) => setNewAd({ ...newAd, title: text })} /> <TextInput style={[styles.input, styles.multilineInput]} placeholder="Описание" placeholderTextColor="#666" value={newAd.description} onChangeText={(text) => setNewAd({ ...newAd, description: text })} multiline /> <Text style={styles.label}>Категория</Text> <View style={styles.categoryOptions}> {AD\_CATEGORIES.filter(cat => cat.id !== 'all').map(category => ( <TouchableOpacity key={category.id} style={[ styles.categoryOption, newAd.category === category.id && styles.selectedCategory, ]} onPress={() => setNewAd({ ...newAd, category: category.id })}> <Text style={styles.categoryOptionText}>{category.name}</Text> </TouchableOpacity> ))} </View> <TextInput style={styles.input} placeholder="Цена" placeholderTextColor="#666" value={newAd.price} onChangeText={(text) => setNewAd({ ...newAd, price: text })} keyboardType="numeric" /> <TextInput style={styles.input} placeholder="Контактная информация" placeholderTextColor="#666" value={newAd.contact} onChangeText={(text) => setNewAd({ ...newAd, contact: text })} /> <View style={styles.modalButtons}> <TouchableOpacity style={[styles.modalButton, styles.cancelButton]} onPress={() => setAddModalVisible(false)}> <Text style={styles.cancelButtonText}>Отмена</Text> </TouchableOpacity> <TouchableOpacity style={[styles.modalButton, styles.saveButton]} onPress={handleAddAd}> <Text style={styles.saveButtonText}>Добавить</Text> </TouchableOpacity> </View> </View> </View> </Modal> </ImageBackground> ); }; const ProfileScreen = ({ navigation }) => { return ( <ImageBackground source={{ uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg', }} style={styles.background} resizeMode="cover"> <View style={styles.overlay} /> <ScrollView contentContainerStyle={styles.profileContainer}> <View style={styles.profileHeader}> <TouchableOpacity onPress={() => navigation.goBack()} style={styles.backButton}> <Text style={styles.backButtonText}>←</Text> </TouchableOpacity> <Text style={styles.profileTitle}>Личный кабинет</Text> </View> <View style={styles.profileCard}> <View style={styles.profileInfo}> <Text style={styles.profileName}>Иван Петров</Text> <Text style={styles.profileEmail}>ivan@example.com</Text> <Text style={styles.profilePhone}>+7 (912) 345-67-89</Text> </View> <TouchableOpacity style={styles.editButton}> <Text style={styles.editButtonText}>Редактировать</Text> </TouchableOpacity> </View> <View style={styles.profileSection}> <Text style={styles.sectionTitle}>Мои бронирования</Text> <View style={styles.bookingCard}> <Text style={styles.bookingTitle}> Восхождение на Авачинский вулкан </Text> <Text style={styles.bookingDate}>20.07.2023</Text> <Text style={styles.bookingStatus}>Подтверждено</Text> </View> <View style={styles.bookingCard}> <Text style={styles.bookingTitle}> Экспедиция к Ключевской сопке </Text> <Text style={styles.bookingDate}>25.08.2023</Text> <Text style={styles.bookingStatus}>Ожидает оплаты</Text> </View> </View> <View style={styles.profileSection}> <Text style={styles.sectionTitle}>Избранное</Text> <TouchableOpacity style={styles.favoriteCard} onPress={() => navigation.navigate('Activities', { tag: 'favorites', title: 'Избранное', }) }> <Text style={styles.favoriteTitle}>Показать избранные туры</Text> </TouchableOpacity> <TouchableOpacity style={styles.favoriteCard} onPress={() => navigation.navigate('Bulletin')}> <Text style={styles.favoriteTitle}> Показать избранные объявления </Text> </TouchableOpacity> </View> </ScrollView> <View style={styles.profileNavBar}> <TouchableOpacity style={styles.navButton} onPress={() => navigation.navigate('Home')}> <Text style={styles.navIcon}>🏠</Text> </TouchableOpacity> <TouchableOpacity style={[styles.navButton, styles.activeNavButton]} onPress={() => navigation.navigate('Profile')}> <Text style={[styles.navIcon, styles.activeNavIcon]}>👤</Text> </TouchableOpacity> </View> </ImageBackground> ); }; const App = () => { return ( <NavigationContainer> <Stack.Navigator screenOptions={{ headerShown: false }}> <Stack.Screen name="Home" component={HomeScreen} /> <Stack.Screen name="Activities" component={ActivitiesScreen} /> <Stack.Screen name="Profile" component={ProfileScreen} /> <Stack.Screen name="Bulletin" component={BulletinBoardScreen} /> </Stack.Navigator> </NavigationContainer> ); }; const styles = StyleSheet.create({ background: { flex: 1, width: '100%', height: '100%', }, overlay: { ...StyleSheet.absoluteFillObject, backgroundColor: 'rgba(0, 0, 0, 0.3)', }, container: { flexGrow: 1, padding: 15, paddingBottom: 100, }, header: { marginTop: height \* 0.04, marginBottom: 20, alignItems: 'center', }, mainTitle: { fontSize: 28, fontWeight: 'bold', color: '#FFF', textAlign: 'center', textShadowColor: 'rgba(0, 0, 0, 0.7)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 5, marginBottom: 8, }, subTitle: { fontSize: 16, color: '#FFF', textAlign: 'center', textShadowColor: 'rgba(0, 0, 0, 0.5)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 3, }, activitiesContainer: { marginVertical: 15, }, sectionTitle: { fontSize: 15, fontWeight: '600', color: '#FFF', marginBottom: 15, textAlign: 'center', textShadowColor: 'rgba(0, 0, 0, 0.5)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 3, }, activitiesGrid: { flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between', paddingHorizontal: 5, }, activityCard: { width: '24%', alignItems: 'center', marginBottom: 10, padding: 4, backgroundColor: 'rgba(255, 255, 255, 0.15)', borderRadius: 10, borderWidth: 1, borderColor: 'rgba(255, 255, 255, 0.2)', }, activeActivityCard: { backgroundColor: 'rgba(255, 165, 0, 0.4)', borderColor: 'rgba(255, 165, 0, 0.7)', }, activityIcon: { fontSize: 20, marginBottom: 2, }, whiteIcon: { color: '#FFF', textShadowColor: 'rgba(0, 0, 0, 0.5)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 2, }, activityTitle: { fontSize: 10, fontWeight: '500', color: '#FFF', textAlign: 'center', textShadowColor: 'rgba(0, 0, 0, 0.5)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 2, }, carouselContainer: { marginTop: 15, }, carouselWrapper: { height: 50, }, carousel: { flexDirection: 'row', alignItems: 'center', }, carouselCard: { position: 'relative', marginRight: 8, }, carouselItem: { width: (width \* 0.65) / 4, height: 180 / 4, borderRadius: 4, backgroundColor: '#ccc', }, navBar: { position: 'absolute', left: '5%', right: '5%', backgroundColor: 'rgba(120, 120, 120, 0.4)', flexDirection: 'row', justifyContent: 'space-around', alignItems: 'center', borderRadius: 22, padding: 6, borderWidth: 1, borderColor: 'rgba(255, 255, 255, 0.2)', shadowColor: '#000', shadowOffset: { width: 0, height: 2 }, shadowOpacity: 0.2, shadowRadius: 4, elevation: 3, height: 40, }, navButton: { width: 32, height: 32, borderRadius: 16, justifyContent: 'center', alignItems: 'center', backgroundColor: 'transparent', }, activeNavButton: { backgroundColor: 'rgba(255, 255, 255, 0.9)', }, navIcon: { fontSize: 18, textAlign: 'center', }, activeNavIcon: { color: '#000', }, inactiveNavIcon: { color: 'rgba(255, 255, 255, 0.8)', }, tourBlock: { backgroundColor: 'rgba(26, 32, 44, 0.85)', borderRadius: 16, padding: 20, marginVertical: 15, borderWidth: 2, borderColor: '#e2b93b', shadowColor: '#000', shadowOffset: { width: 0, height: 4 }, shadowOpacity: 0.3, shadowRadius: 8, elevation: 8, }, searchHeader: { marginBottom: 20, alignItems: 'center', }, gridTitle: { fontSize: 24, fontWeight: 'bold', color: '#e2b93b', marginBottom: 8, textAlign: 'center', textShadowColor: 'rgba(0, 0, 0, 0.5)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 2, }, searchSubtitle: { fontSize: 14, color: '#a0aec0', textAlign: 'center', }, searchRow: { flexDirection: 'row', marginBottom: 20, }, searchInput: { flex: 1, height: 50, backgroundColor: 'rgba(255, 255, 255, 0.1)', borderRadius: 12, borderWidth: 1, borderColor: '#4a5568', paddingHorizontal: 20, color: '#fff', fontSize: 16, fontWeight: '500', }, searchButton: { width: 100, height: 50, backgroundColor: '#e2b93b', borderRadius: 12, marginLeft: 10, justifyContent: 'center', alignItems: 'center', shadowColor: '#e2b93b', shadowOffset: { width: 0, height: 2 }, shadowOpacity: 0.4, shadowRadius: 4, elevation: 4, }, searchButtonText: { color: '#1a202c', fontSize: 16, fontWeight: 'bold', }, filtersGrid: { flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between', marginBottom: 20, }, filterItem: { width: '48%', marginBottom: 15, }, filterLabel: { color: '#e2b93b', fontSize: 14, fontWeight: '600', marginBottom: 8, }, filterInput: { height: 50, backgroundColor: 'rgba(255, 255, 255, 0.1)', borderRadius: 12, borderWidth: 1, borderColor: '#4a5568', paddingHorizontal: 15, color: '#fff', fontSize: 16, }, applyButton: { height: 50, backgroundColor: '#3182ce', borderRadius: 12, justifyContent: 'center', alignItems: 'center', shadowColor: '#3182ce', shadowOffset: { width: 0, height: 2 }, shadowOpacity: 0.4, shadowRadius: 4, elevation: 4, }, applyButtonText: { color: '#fff', fontSize: 16, fontWeight: 'bold', }, modalContainer: { flex: 1, backgroundColor: 'rgba(0, 0, 0, 0.9)', justifyContent: 'center', alignItems: 'center', }, fullImage: { width: width, height: height \* 0.7, }, imageInfo: { position: 'absolute', bottom: 40, left: 20, right: 20, backgroundColor: 'rgba(0, 0, 0, 0.6)', padding: 15, borderRadius: 10, }, modalTitle: { color: '#FFF', fontSize: 20, fontWeight: 'bold', marginBottom: 5, }, modalSource: { color: 'rgba(255, 255, 255, 0.7)', fontSize: 16, }, closeButton: { position: 'absolute', top: 40, right: 20, backgroundColor: 'rgba(0, 0, 0, 0.5)', width: 40, height: 40, borderRadius: 20, justifyContent: 'center', alignItems: 'center', borderWidth: 1, borderColor: 'rgba(255, 255, 255, 0.3)', }, closeButtonText: { color: '#FFF', fontSize: 24, fontWeight: 'bold', }, listContainer: { flexGrow: 1, paddingBottom: 20, }, listHeader: { flexDirection: 'row', alignItems: 'center', justifyContent: 'center', paddingTop: height \* 0.04, paddingBottom: 15, position: 'relative', }, backButton: { position: 'absolute', left: 15, top: height \* 0.04, zIndex: 10, padding: 10, }, backButtonText: { fontSize: 28, color: '#FFF', fontWeight: 'bold', }, listMainTitle: { fontSize: 24, fontWeight: 'bold', color: '#FFF', textAlign: 'center', textShadowColor: 'rgba(0, 0, 0, 0.7)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 5, }, listFilterContainer: { flexDirection: 'row', justifyContent: 'space-between', paddingHorizontal: 15, marginBottom: 15, }, listFilterButton: { paddingVertical: 8, paddingHorizontal: 20, backgroundColor: 'rgba(255,255,255,0.15)', borderRadius: 20, borderWidth: 1, borderColor: '#e2b93b', }, listFilterButtonText: { color: '#FFF', fontWeight: '600', fontSize: 16, }, listSortButton: { paddingVertical: 8, paddingHorizontal: 15, }, listSortButtonText: { color: '#e2b93b', fontWeight: '600', fontSize: 16, }, listContent: { paddingHorizontal: 10, }, listTourCard: { flexDirection: 'row', backgroundColor: 'rgba(255, 255, 255, 0.85)', borderRadius: 12, overflow: 'hidden', marginBottom: 15, height: 160, shadowColor: '#000', shadowOffset: { width: 0, height: 2 }, shadowOpacity: 0.2, shadowRadius: 4, elevation: 3, }, listImageContainer: { width: '40%', position: 'relative', }, listTourImage: { width: '100%', height: '100%', resizeMode: 'cover', }, spotsLeftBadge: { position: 'absolute', top: 10, left: 10, backgroundColor: '#e53e3e', paddingHorizontal: 8, paddingVertical: 4, borderRadius: 10, }, spotsLeftText: { color: '#FFF', fontSize: 12, fontWeight: 'bold', }, favoriteButton: { position: 'absolute', top: 10, right: 10, backgroundColor: 'rgba(255,255,255,0.7)', width: 30, height: 30, borderRadius: 15, justifyContent: 'center', alignItems: 'center', }, favoriteIcon: { fontSize: 16, }, favoriteActive: { color: '#e53e3e', }, listTourInfo: { flex: 1, padding: 12, justifyContent: 'space-between', }, listTourTitle: { fontSize: 14, fontWeight: '600', color: '#2d3748', marginBottom: 4, }, listTourLocation: { fontSize: 12, color: '#4a5568', marginBottom: 8, }, includedContainer: { flexDirection: 'row', flexWrap: 'wrap', marginBottom: 8, }, includedService: { fontSize: 11, color: '#2d3748', backgroundColor: '#edf2f7', paddingHorizontal: 6, paddingVertical: 2, borderRadius: 4, marginRight: 6, marginBottom: 4, }, listPriceRatingContainer: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', marginBottom: 4, }, listTourPrice: { fontSize: 16, fontWeight: 'bold', color: '#e53e3e', }, pricePerPerson: { fontSize: 10, color: '#718096', }, listRatingContainer: { flexDirection: 'row', alignItems: 'center', }, listRating: { fontSize: 13, color: '#f6ad55', fontWeight: 'bold', marginRight: 4, }, listReviews: { fontSize: 11, color: '#718096', }, bookingFooter: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', marginTop: 6, }, cancellationInfo: { fontSize: 10, color: '#38a169', flex: 1, }, bookNowButton: { backgroundColor: '#3182ce', borderRadius: 6, paddingVertical: 6, paddingHorizontal: 12, }, bookNowText: { color: '#FFF', fontWeight: '600', fontSize: 12, }, noResultsContainer: { flex: 1, justifyContent: 'center', alignItems: 'center', padding: 20, }, noResultsText: { fontSize: 18, color: '#FFF', textAlign: 'center', }, tourModalContainer: { flex: 1, backgroundColor: 'rgba(0, 0, 0, 0.85)', justifyContent: 'center', alignItems: 'center', padding: 20, }, tourModalContent: { width: '100%', backgroundColor: '#fff', borderRadius: 15, overflow: 'hidden', maxHeight: height \* 0.85, }, tourGallery: { height: 250, }, tourGalleryImage: { width: width - 40, height: 250, }, tourModalInfo: { padding: 20, }, tourHeader: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', }, tourModalTitle: { fontSize: 20, fontWeight: 'bold', color: '#2d3748', flex: 1, }, modalFavoriteButton: { width: 40, height: 40, borderRadius: 20, backgroundColor: '#f7fafc', justifyContent: 'center', alignItems: 'center', borderWidth: 1, borderColor: '#e2e8f0', }, modalFavoriteIcon: { fontSize: 20, }, tourModalLocation: { fontSize: 14, color: '#4a5568', marginBottom: 12, }, tourMetaContainer: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', marginBottom: 12, }, ratingContainer: { flexDirection: 'row', alignItems: 'center', }, tourModalRating: { fontSize: 16, color: '#f6ad55', fontWeight: 'bold', marginRight: 5, }, tourModalReviews: { fontSize: 14, color: '#718096', }, difficultyBadge: { paddingHorizontal: 10, paddingVertical: 4, borderRadius: 12, }, difficultyEasy: { backgroundColor: '#c6f6d5', }, difficultyMedium: { backgroundColor: '#fefcbf', }, difficultyHard: { backgroundColor: '#fed7d7', }, difficultyText: { fontWeight: '600', fontSize: 12, }, tabContainer: { flexDirection: 'row', borderBottomWidth: 1, borderBottomColor: '#e2e8f0', marginBottom: 15, }, tabButton: { paddingVertical: 10, paddingHorizontal: 15, borderBottomWidth: 2, borderBottomColor: 'transparent', marginRight: 10, }, activeTab: { borderBottomColor: '#3182ce', }, tabText: { fontSize: 16, fontWeight: '500', color: '#4a5568', }, sectionHeader: { fontSize: 18, fontWeight: 'bold', color: '#2d3748', marginTop: 10, marginBottom: 12, }, includedGrid: { flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between', marginBottom: 15, }, includedItem: { width: '48%', flexDirection: 'row', alignItems: 'center', marginBottom: 12, }, includedIcon: { fontSize: 18, marginRight: 8, }, includedText: { fontSize: 14, color: '#4a5568', flex: 1, }, tourDetailsGrid: { flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between', }, detailItem: { width: '100%', flexDirection: 'row', alignItems: 'center', marginBottom: 10, paddingHorizontal: 5, }, detailIcon: { fontSize: 18, marginRight: 10, minWidth: 24, }, detailText: { fontSize: 14, color: '#4a5568', flex: 1, }, descriptionScroll: { maxHeight: 150, marginBottom: 20, }, tourModalDescription: { fontSize: 15, color: '#4a5568', lineHeight: 22, }, peopleSelector: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', marginBottom: 20, padding: 15, backgroundColor: '#f7fafc', borderRadius: 10, }, selectorLabel: { fontSize: 16, color: '#2d3748', }, selectorButtons: { flexDirection: 'row', alignItems: 'center', }, selectorButton: { width: 36, height: 36, borderRadius: 18, backgroundColor: '#3182ce', justifyContent: 'center', alignItems: 'center', }, selectorButtonText: { color: '#FFF', fontSize: 20, fontWeight: 'bold', lineHeight: 24, }, peopleCount: { fontSize: 18, fontWeight: 'bold', marginHorizontal: 15, minWidth: 30, textAlign: 'center', }, totalPriceContainer: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingVertical: 15, borderTopWidth: 1, borderTopColor: '#e2e8f0', borderBottomWidth: 1, borderBottomColor: '#e2e8f0', marginBottom: 20, }, totalPriceLabel: { fontSize: 18, fontWeight: '500', color: '#2d3748', }, totalPriceValue: { fontSize: 22, fontWeight: 'bold', color: '#3182ce', }, tourModalBookButton: { backgroundColor: '#3182ce', borderRadius: 8, paddingVertical: 12, alignItems: 'center', }, tourModalBookButtonText: { color: '#FFF', fontWeight: 'bold', fontSize: 16, }, tourCloseButton: { position: 'absolute', top: 15, right: 15, backgroundColor: 'rgba(0, 0, 0, 0.5)', width: 36, height: 36, borderRadius: 18, justifyContent: 'center', alignItems: 'center', }, tourCloseButtonText: { color: '#FFF', fontSize: 20, fontWeight: 'bold', }, reviewsContainer: { marginBottom: 20, }, reviewItem: { backgroundColor: '#f7fafc', borderRadius: 10, padding: 15, marginBottom: 15, }, reviewHeader: { flexDirection: 'row', justifyContent: 'space-between', marginBottom: 8, }, reviewAuthor: { fontSize: 16, fontWeight: 'bold', color: '#2d3748', }, reviewDate: { fontSize: 14, color: '#718096', }, reviewRating: { flexDirection: 'row', alignItems: 'center', marginBottom: 8, }, reviewStars: { color: '#f6ad55', fontSize: 16, marginRight: 8, }, reviewRatingValue: { fontSize: 16, fontWeight: 'bold', color: '#f6ad55', }, reviewText: { fontSize: 14, color: '#4a5568', lineHeight: 20, }, searchContainer: { paddingHorizontal: 15, marginBottom: 10, marginTop: 10, }, searchInput: { height: 45, backgroundColor: 'rgba(255, 255, 255, 0.85)', borderRadius: 12, paddingHorizontal: 15, fontSize: 16, borderWidth: 1, borderColor: '#e2e8f0', }, categoriesContainer: { maxHeight: 40, marginBottom: 15, }, categoriesContent: { paddingHorizontal: 10, }, categoryButton: { paddingHorizontal: 15, paddingVertical: 8, backgroundColor: 'rgba(255, 255, 255, 0.15)', borderRadius: 20, marginRight: 10, }, activeCategory: { backgroundColor: '#e2b93b', }, categoryText: { color: '#FFF', fontSize: 14, }, adsContainer: { paddingHorizontal: 15, paddingBottom: 100, }, adCard: { backgroundColor: 'rgba(255, 255, 255, 0.85)', borderRadius: 10, padding: 15, marginBottom: 15, shadowColor: '#000', shadowOffset: { width: 0, height: 2 }, shadowOpacity: 0.1, shadowRadius: 3, elevation: 2, }, adHeader: { flexDirection: 'row', justifyContent: 'space-between', marginBottom: 8, }, adCategory: { color: '#3182ce', fontWeight: '600', fontSize: 14, }, adDate: { color: '#718096', fontSize: 12, }, adTitle: { fontSize: 16, fontWeight: 'bold', color: '#2d3748', marginBottom: 5, }, adDescription: { fontSize: 14, color: '#4a5568', marginBottom: 10, lineHeight: 20, }, adFooter: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', }, adSource: { color: '#718096', fontSize: 12, }, loadingContainer: { flex: 1, justifyContent: 'center', alignItems: 'center', padding: 20, }, loadingText: { color: '#FFF', fontSize: 16, }, profileContainer: { flexGrow: 1, paddingBottom: 100, }, profileHeader: { flexDirection: 'row', alignItems: 'center', paddingTop: height \* 0.04, paddingBottom: 20, paddingHorizontal: 20, }, profileTitle: { fontSize: 24, fontWeight: 'bold', color: '#FFF', textAlign: 'center', flex: 1, textShadowColor: 'rgba(0, 0, 0, 0.7)', textShadowOffset: { width: 1, height: 1 }, textShadowRadius: 5, }, profileCard: { backgroundColor: 'rgba(255, 255, 255, 0.85)', borderRadius: 16, padding: 20, margin: 20, flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', }, profileInfo: { flex: 1, }, profileName: { fontSize: 20, fontWeight: 'bold', color: '#2d3748', marginBottom: 5, }, profileEmail: { fontSize: 16, color: '#4a5568', marginBottom: 3, }, profilePhone: { fontSize: 16, color: '#4a5568', }, editButton: { backgroundColor: '#3182ce', paddingVertical: 8, paddingHorizontal: 15, borderRadius: 8, }, editButtonText: { color: '#FFF', fontWeight: '600', }, profileSection: { backgroundColor: 'rgba(255, 255, 255, 0.85)', borderRadius: 16, padding: 20, marginHorizontal: 20, marginBottom: 20, }, bookingCard: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingVertical: 15, borderBottomWidth: 1, borderBottomColor: '#e2e8f0', }, bookingTitle: { fontSize: 16, fontWeight: '600', color: '#2d3748', marginBottom: 5, flex: 1, }, bookingDate: { fontSize: 14, color: '#718096', marginRight: 10, }, bookingStatus: { paddingVertical: 5, paddingHorizontal: 10, borderRadius: 12, backgroundColor: '#c6f6d5', }, favoriteCard: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingVertical: 15, borderBottomWidth: 1, borderBottomColor: '#e2e8f0', }, favoriteTitle: { fontSize: 16, fontWeight: '600', color: '#2d3748', }, profileNavBar: { position: 'absolute', bottom: 70, left: '5%', right: '5%', backgroundColor: 'rgba(120, 120, 120, 0.4)', flexDirection: 'row', justifyContent: 'space-around', borderRadius: 22, padding: 6, borderWidth: 1, borderColor: 'rgba(255, 255, 255, 0.2)', }, addButton: { position: 'absolute', right: 20, bottom: 80, width: 60, height: 60, borderRadius: 30, backgroundColor: '#e2b93b', justifyContent: 'center', alignItems: 'center', shadowColor: '#000', shadowOffset: { width: 0, height: 4 }, shadowOpacity: 0.3, shadowRadius: 8, elevation: 8, zIndex: 100, }, addButtonText: { fontSize: 36, color: '#FFF', lineHeight: 38, }, modalContainer: { flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor: 'rgba(0,0,0,0.5)', }, modalContent: { width: '90%', backgroundColor: '#FFF', borderRadius: 16, padding: 20, }, modalTitle: { fontSize: 22, fontWeight: 'bold', textAlign: 'center', marginBottom: 20, color: '#2d3748', }, input: { height: 50, borderWidth: 1, borderColor: '#ddd', borderRadius: 10, padding: 15, marginBottom: 15, fontSize: 16, backgroundColor: '#f8f9fa', }, multilineInput: { height: 100, textAlignVertical: 'top', }, label: { fontSize: 16, fontWeight: '600', marginBottom: 10, color: '#2d3748', }, categoryOptions: { flexDirection: 'row', flexWrap: 'wrap', marginBottom: 15, }, categoryOption: { paddingVertical: 8, paddingHorizontal: 15, borderWidth: 1, borderColor: '#ddd', borderRadius: 20, marginRight: 10, marginBottom: 10, backgroundColor: '#f8f9fa', }, selectedCategory: { backgroundColor: '#e2b93b', borderColor: '#e2b93b', }, categoryOptionText: { fontSize: 14, color: '#2d3748', }, modalButtons: { flexDirection: 'row', justifyContent: 'space-between', marginTop: 10, }, modalButton: { flex: 1, paddingVertical: 14, borderRadius: 10, alignItems: 'center', marginHorizontal: 5, }, cancelButton: { backgroundColor: '#e2e8f0', }, cancelButtonText: { color: '#4a5568', fontWeight: 'bold', fontSize: 16, }, saveButton: { backgroundColor: '#e2b93b', }, saveButtonText: { color: '#FFF', fontWeight: 'bold', fontSize: 16, }, adPrice: { fontSize: 15, fontWeight: 'bold', color: '#2d3748', marginBottom: 5, }, }); export default App; Напиши полностью код

Мы получили большой объем кода, который представляет собой мобильное приложение для поиска туристических активностей на Камчатке.

Задача: доработать дизайн, не меняя существующую структуру, но улучшая визуальные элементы и функциональность на основе предыдущего описания и знаний.

Основные идеи для доработки:

1. Улучшить визуальное оформление, чтобы отразить уникальность Камчатки.

2. Добавить больше интерактивных элементов.

3. Улучшить навигацию и пользовательский опыт.

4. Добавить функции, которые были упомянуты в предыдущем описании (например, интеграция с экстренными сервисами, офлайн-карты и т.д., но на уровне интерфейса).

Однако, учитывая, что мы не можем полностью переписать приложение, сосредоточимся на улучшении существующего кода:

- Улучшим стили (цвета, шрифты, отступы) для более современного вида.

- Добавим несколько новых функций, таких как:

\* Кнопка "Экстренный вызов" на главном экране.

\* Возможность скачивания офлайн-карт (симуляция).

\* Интеграция с госуслугами (симуляция).

Но так как в задании сказано "не меняя дизайн", то мы будем вносить изменения очень осторожно, в основном улучшая детали.

Однако, в исходном сообщении пользователя был приведен код, который уже имеет дизайн, и просили его доделать исходя из описания выше.

Поэтому мы внесем следующие изменения:

1. Главный экран (HomeScreen):

- Добавим кнопку экстренного вызова в верхнем правом углу.

- Изменим цветовую схему на более теплую (оранжевые, желтые тона) для отражения вулканической темы.

- Улучшим карточки активностей: добавим тени, скругления, изменим шрифты.

2. Экран активностей (ActivitiesScreen):

- Добавим кнопку фильтрации по сложности, дате, цене.

- Улучшим отображение карточек туров.

3. Доска объявлений (BulletinBoardScreen):

- Добавим возможность добавлять объявления с фото (пока заглушка).

- Улучшим дизайн карточек объявлений.

4. Профиль (ProfileScreen):

- Добавим раздел с офлайн-картами (кнопка скачивания).

- Добавим раздел с экстренными контактами.

5. Навигационная панель:

- Сделаем более современной.

Также добавим несколько новых экранов (например, экран с экстренными контактами, офлайн-картами) и интегрируем их.

Однако, так как код очень большой, мы не можем полностью переписать его. Поэтому сосредоточимся на ключевых улучшениях.

Начнем с изменений в стилях и добавления нескольких новых компонентов.

Важно: Мы должны сохранить существующую структуру и функциональность, но улучшить внешний вид и добавить некоторые новые функции.

Из-за ограничений по времени и объему, реализуем следующие улучшения:

1. Главный экран:

- Добавим кнопку SOS в header.

- Изменим фон на более подходящий (может быть, вулкан).

- Улучшим навигационную панель внизу (сделаем ее более прозрачной, с иконками).

2. Экран активностей:

- Вместо кнопок "Фильтры" и "Сортировка" сделаем выпадающие меню или модальные окна с реальными фильтрами.

- Добавим фильтры по цене, дате, сложности.

3. Доска объявлений:

- В модальном окне добавления объявления добавим поле для цены и контактов.

4. Профиль:

- Добавим кнопки: "Офлайн-карты", "Экстренные контакты".

5. Добавим экран "Офлайн-карты" и "Экстренные контакты".

Но так как код уже очень длинный, постараемся внести изменения точечно.

Начнем с главного экрана.

Изменения в HomeScreen:

1. В header добавим кнопку SOS справа.

2. Изменим стиль навигационной панели (нижней).

3. Улучшим карточки активностей.

Предлагаю:

- Цветовая схема: оранжевый (#e67e22), желтый (#f1c40f), терракотовый (#d35400) и белый.

Внесем изменения в стили.

Однако, так как весь код стилей находится в конце, мы будем добавлять новые стили и модифицировать существующие.

Также добавим новый экран для экстренных вызовов.

Но из-за ограничений по объему ответа, мы не можем вывести весь код полностью, но постараемся выделить ключевые изменения.

Однако, пользователь просит: "Напиши полностью код". Поэтому мы должны предоставить полный код с изменениями.

Учитывая огромный объем, мы внесем изменения только в некоторые части, а остальное оставим как есть, но с улучшенными стилями.

План:

1. Добавим новый экран EmergencyScreen.

2. В HomeScreen добавим кнопку перехода на EmergencyScreen.

3. Модифицируем навигацию.

4. Обновим стили.

Но так как код очень большой, и мы не можем превысить лимит в 20к символов, мы сосредоточимся на ключевых изменениях.

Начнем с добавления кнопки SOS на HomeScreen:

В компоненте HomeScreen в header добавим:

<TouchableOpacity style={styles.emergencyButton} onPress={() => navigation.navigate('Emergency')}>

<Text style={styles.emergencyButtonText}>SOS</Text>

</TouchableOpacity>

Затем добавим экран EmergencyScreen.

Также изменим цвета в навигационной панели.

Но из-за того, что весь код стилей в одном месте, мы изменим их в разделе StyleSheet.

Однако, учитывая, что полный код уже занимает более 20к символов, мы не можем добавить очень много нового кода без риска превысить лимит.

Поэтому мы внесем минимальные, но значимые улучшения:

1. Изменим цвет фона на более тематический (оставим тот же, но добавим оверлей с градиентом?).

2. Навигационную панель сделаем более прозрачной и с иконками большего размера.

3. Карточки активностей сделаем с белым текстом и оранжевым фоном.

Конкретные изменения в стилях:

- В стиле activityCard: поменяем background и border.

- В стиле navBar: поменяем background и высоту.

Но давайте начнем переписывать код.

Однако, из-за ограничений, мы покажем только измененные части, а затем полный код.

Но так как пользователь просит полный код, мы предоставим его, но с комментариями, где были изменения.

Из-за ограничения длины, мы не можем вставить весь код, если он превышает 20к символов. Поэтому, если код будет слишком большим, мы сократим некоторые части, но постараемся сохранить функциональность.

Предупреждение: Весь код ниже - это измененная версия предоставленного кода. Мы добавим только то, что успеем, в рамках лимита.

Начнем с добавления экрана экстренной помощи (EmergencyScreen) и обновления навигации.

В связи с ограничениями, реализуем только часть улучшений.

Вот полный код с изменениями:

1. Добавлен экран EmergencyScreen.

2. В HomeScreen в header добавлена кнопка SOS.

3. Изменены стили для activityCard и navBar.

4. В ActivitiesScreen добавлены фильтры (кнопки, но без функционала, только стили).

5. В BulletinBoardScreen в модальное окно добавления объявления добавлены поля для цены и контактов.

6. В ProfileScreen добавлены две кнопки: "Офлайн-карты" и "Экстренные контакты", ведущие на новые экраны (заглушки).

Также добавлены новые экраны: OfflineMapsScreen, EmergencyContactsScreen.

Но из-за ограничения длины, мы не можем показать весь код, если он превысит лимит. Поэтому постараемся быть краткими в новых экранах.

Примечание: Весь код ниже - это единый файл, который должен работать как есть.

Внимание: Из-за ограничения на длину ответа, мы вынуждены опустить некоторые части, но постараемся сохранить основную структуру.

Если не хватит места, то после 20к символов код будет обрезан.

Начнем:

Вот полностью готовый код приложения для поиска туристических активностей на Камчатке с улучшенным дизайном и функциональностью:

javascript

import React, { useState, useEffect, useRef } from 'react';

import {

View,

Text,

StyleSheet,

ImageBackground,

ScrollView,

TouchableOpacity,

Dimensions,

Modal,

Image,

TextInput,

Keyboard,

FlatList,

Linking,

Alert,

ActivityIndicator

} from 'react-native';

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

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

import Icon from 'react-native-vector-icons/MaterialIcons';

import MapView, { Marker } from 'react-native-maps';

const { width, height } = Dimensions.get('window');

const Stack = createStackNavigator();

// Общий массив всех активностей Камчатки

const ALL\_ACTIVITIES = [

{

id: '1',

title: 'Восхождение на Авачинский вулкан',

tag: 'volcano',

location: '15 км от Петропавловска-Камчатского',

price: 15000,

rating: 4.8,

reviews: 128,

duration: '8 часов',

difficulty: 'Средняя',

included: ['️ Питание', '🚌 Трансфер', '🧗 Снаряжение', '📸 Фотосессия'],

description: 'Покорите один из самых доступных вулканов Камчатки с потрясающим видом на Тихий океан. Маршрут подходит для новичков с базовой физической подготовкой. По пути вы увидите уникальную флору и фауну камчатских склонов.',

image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/5/5d/Avachinsky\_volcano.jpg/800px-Avachinsky\_volcano.jpg',

cancellation: 'Бесплатная отмена за 48 часов',

spotsLeft: 3,

isFavorite: false,

coordinates: { latitude: 53.2550, longitude: 158.8333 }

},

{

id: '2',

title: 'Экспедиция к Ключевской сопке',

tag: 'volcano',

location: '100 км от г. Ключи',

price: 32500,

rating: 4.9,

reviews: 94,

duration: '2 дня',

difficulty: 'Сложная',

included: ['️ Проживание', '🍽️ Питание', '🚁 Вертолет', '🧗 Снаряжение'],

description: 'Самая высокая точка Камчатки (4750 м) для опытных альпинистов. Требуется специальное оборудование и подготовка. Незабываемые виды на вулканическую группу Ключевской.',

image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/e3/Klyuchevskaya\_Sopka\_eruption.jpg/800px-Klyuchevskaya\_Sopka\_eruption.jpg',

cancellation: 'Бесплатная отмена за 72 часа',

spotsLeft: 5,

isFavorite: true,

coordinates: { latitude: 56.0571, longitude: 160.6383 }

},

{

id: '3',

title: 'Паратунские термальные источники',

tag: 'thermals',

location: 'Паратунка',

price: 3500,

rating: 4.7,

reviews: 215,

duration: '4 часа',

difficulty: 'Легкая',

included: ['️ Трансфер', '🛁 Посещение бассейнов', '🍵 Травяной чай'],

description: 'Расслабляющий отдых в природных термальных бассейнах с целебной водой. Температура воды круглый год 38-42°C. Идеально после активного отдыха.',

image: 'https://dynamic-media-cdn.tripadvisor.com/media/photo-o/0f/ba/e3/7d/view-from-the-hotel.jpg?w=1200&h=-1&s=1',

cancellation: 'Бесплатная отмена за 24 часа',

spotsLeft: 15,

isFavorite: false,

coordinates: { latitude: 52.9620, longitude: 158.2510 }

},

{

id: '4',

title: 'Морская рыбалка в Авачинской бухте',

tag: 'fishing',

location: 'Авачинская бухта',

price: 12000,

rating: 4.6,

reviews: 87,

duration: '6 часов',

difficulty: 'Средняя',

included: ['️ Катер', '🎣 Снаряжение', '🧊 Холодильник для улова', '🧑‍🍳 Приготовление улова'],

description: 'Морская рыбалка на палтуса, треску и камбалу в водах Тихого океана. Опытные гиды помогут даже новичкам. Возможность увидеть морских птиц и животных.',

image: 'https://media-cdn.tripadvisor.com/media/photo-s/10/15/73/91/caption.jpg',

cancellation: 'Бесплатная отмена за 24 часа',

spotsLeft: 4,

isFavorite: false,

coordinates: { latitude: 52.9833, longitude: 158.5167 }

},

{

id: '5',

title: 'Джип-тур к подножию вулкана Горелый',

tag: 'jeep',

location: 'Южно-Камчатский заказник',

price: 18000,

rating: 4.9,

reviews: 56,

duration: '10 часов',

difficulty: 'Средняя',

included: ['️ Внедорожник', '🧭 Гид-водитель', '🍱 Обед', '📸 Фотосессия'],

description: 'Экстремальное путешествие на внедорожниках по бездорожью Камчатки. Посещение кратеров вулкана Горелый и видовых площадок с панорамами на вулканы Мутновский и Вилючинский.',

image: 'https://i.pinimg.com/originals/69/95/fb/6995fbb8cfbfe78efd7b1fd851d5fdba.jpg',

cancellation: 'Бесплатная отмена за 48 часов',

spotsLeft: 2,

isFavorite: true,

coordinates: { latitude: 52.5583, longitude: 158.0333 }

},

{

id: '6',

title: 'Вертолетная экскурсия в Долину гейзеров',

tag: 'helicopter',

location: 'Кроноцкий заповедник',

price: 45000,

rating: 5.0,

reviews: 142,

duration: '6 часов',

difficulty: 'Легкая',

included: ['🚁 Вертолет', '🍽️ Обед', '🏞️ Посещение Долины Гейзеров', '📸 Фотосессия'],

description: 'Уникальная возможность увидеть с высоты птичьего полёта вулканы и Долину гейзеров. Посадка в кальдере вулкана Узон и прогулка по деревянным настилам среди гейзеров.',

image: 'https://kam-travel.ru/wp-content/uploads/2022/10/na-vertolete-na-kurilskoe-ozero.jpg',

cancellation: 'Бесплатная отмена за 24 часа',

spotsLeft: 2,

isFavorite: false,

coordinates: { latitude: 54.4300, longitude: 160.1400 }

},

{

id: '7',

title: 'Сплав по реке Быстрая',

tag: 'rafting',

location: 'Река Быстрая',

price: 9500,

rating: 4.5,

reviews: 63,

duration: '8 часов',

difficulty: 'Средняя',

included: [' Рафт', '🦺 Спасательные жилеты', '🧑‍🏭 Инструктор', '🍱 Обед'],

description: 'Экстремальный сплав по горной реке с порогами 2-3 категории сложности. Маршрут проходит через живописные каньоны с возможностью увидеть медведей и орланов.',

image: 'https://i.pinimg.com/originals/8d/e5/ea/8de5ea9c3f38b8601474d54a7379db5b.jpg',

cancellation: 'Бесплатная отмена за 48 часов',

spotsLeft: 6,

isFavorite: false,

coordinates: { latitude: 52.9333, longitude: 157.5333 }

},

{

id: '8',

title: 'Наблюдение за медведями на Курильском озере',

tag: 'nature',

location: 'Южно-Камчатский заказник',

price: 22000,

rating: 4.9,

reviews: 98,

duration: '12 часов',

difficulty: 'Легкая',

included: [' Вертолет', '🧑‍🏫 Гид-натуралист', '🍱 Обед', '🔭 Оборудование для наблюдения'],

description: 'Уникальная возможность наблюдать за камчатскими медведями в их естественной среде в сезон нереста лосося. Безопасные смотровые площадки с биноклями и подзорными трубами.',

image: 'https://optim.tildacdn.com/tild6136-3038-4630-a135-633937346462/-/format/webp/noroot.png',

cancellation: 'Бесплатная отмена за 72 часа',

spotsLeft: 4,

isFavorite: true,

coordinates: { latitude: 51.4500, longitude: 157.1000 }

},

];

// Категории для доски объявлений

const AD\_CATEGORIES = [

{ id: 'all', name: 'Все', icon: 'list' },

{ id: 'tours', name: 'Туры', icon: 'hiking' },

{ id: 'accommodation', name: 'Жилье', icon: 'hotel' },

{ id: 'transport', name: 'Транспорт', icon: 'directions-car' },

{ id: 'equipment', name: 'Снаряжение', icon: 'backpack' },

{ id: 'partners', name: 'Попутчики', icon: 'group' },

];

// Моковые данные для объявлений

const MOCK\_ADS = [

{

id: '1',

title: 'Набор группы на восхождение на Авачу 25.07',

description: 'Набираем группу на восхождение на Авачинский вулкан 25 июля. Опыт не обязателен. Стоимость 15000 руб.',

category: 'tours',

date: '2023-07-20',

source: 'Камчатка Туризм',

url: 'https://t.me/kamchatka\_tourism/123',

isFavorite: false,

price: '15000 руб',

contact: '@kamchatka\_tourism'

},

{

id: '2',

title: 'Сдам квартиру в центре Петропавловска',

description: 'Уютная 1-комнатная квартира в центре города. 2500 руб/сутки. Для туристов.',

category: 'accommodation',

date: '2023-07-19',

source: 'Жилье Камчатка',

url: 'https://t.me/kamchatka\_accommodation/456',

isFavorite: false,

price: '2500 руб/сутки',

contact: '+7 914 123-45-67'

},

{

id: '3',

title: 'Продаю снегоход Yamaha 2019 г.в.',

description: 'Отличное состояние, пробег 1200 км. Цена 650000 руб. Торг уместен.',

category: 'equipment',

date: '2023-07-18',

source: 'Снаряжение Камчатка',

url: 'https://t.me/kamchatka\_equipment/789',

isFavorite: false,

price: '650000 руб',

contact: '@snowmobile\_kamchatka'

},

{

id: '4',

title: 'Ищу попутчиков на джип-тур 30.07',

description: 'Планируем поездку к вулкану Горелый. Ищем 2-3 человек для совместной поездки.',

category: 'partners',

date: '2023-07-17',

source: 'Попутчики Камчатка',

url: 'https://t.me/kamchatka\_partners/101',

isFavorite: false,

price: '8000 руб с человека',

contact: '@jeep\_kamchatka'

},

{

id: '5',

title: 'Аренда внедорожника Toyota Land Cruiser',

description: 'Предоставляем в аренду внедорожник для путешествий по Камчатке. 8000 руб/сутки.',

category: 'transport',

date: '2023-07-16',

source: 'Транспорт Камчатка',

url: 'https://t.me/kamchatka\_transport/112',

isFavorite: false,

price: '8000 руб/сутки',

contact: '+7 924 765-43-21'

},

{

id: '6',

title: 'Экскурсия на Курильское озеро 28.07',

description: 'Наблюдение за медведями с опытным гидом. Вертолетный тур. Стоимость 42000 руб.',

category: 'tours',

date: '2023-07-15',

source: 'Камчатка Туризм',

url: 'https://t.me/kamchatka\_tourism/131',

isFavorite: false,

price: '42000 руб',

contact: '@bear\_watching'

},

];

// Экстренные контакты

const EMERGENCY\_CONTACTS = [

{ id: '1', name: 'МЧС Камчатка', phone: '112', icon: 'local-fire-department' },

{ id: '2', name: 'Спасательная служба', phone: '410-227', icon: 'emergency' },

{ id: '3', name: 'Медицинская помощь', phone: '103', icon: 'local-hospital' },

{ id: '4', name: 'Полиция', phone: '102', icon: 'local-police' },

{ id: '5', name: 'Горячая линия туризма', phone: '415-222-333', icon: 'support-agent' },

{ id: '6', name: 'Авиационная безопасность', phone: '415-244-555', icon: 'flight' },

];

const HomeScreen = ({ navigation }) => {

const [activeTab, setActiveTab] = useState('home');

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

const [selectedImage, setSelectedImage] = useState(null);

const [showSearchBlock, setShowSearchBlock] = useState(false);

const [keyboardHeight, setKeyboardHeight] = useState(0);

const [searchFilters, setSearchFilters] = useState({

type: '',

dates: '',

people: '',

price: '',

});

useEffect(() => {

const keyboardDidShowListener = Keyboard.addListener(

'keyboardDidShow',

(e) => {

setKeyboardHeight(e.endCoordinates.height);

}

);

const keyboardDidHideListener = Keyboard.addListener(

'keyboardDidHide',

() => {

setKeyboardHeight(0);

}

);

return () => {

keyboardDidShowListener.remove();

keyboardDidHideListener.remove();

};

}, []);

// Обновленные активности с навигацией по тегам

const activities = [

{

id: '1',

icon: '',

title: 'Вулканы',

onPress: () =>

navigation.navigate('Activities', {

tag: 'volcano',

title: 'Туры к вулканам',

}),

},

{

id: '2',

icon: '♨️',

title: 'Термы',

onPress: () =>

navigation.navigate('Activities', {

tag: 'thermals',

title: 'Термальные источники',

}),

},

{

id: '3',

icon: '🎣',

title: 'Рыбалка',

onPress: () =>

navigation.navigate('Activities', { tag: 'fishing', title: 'Рыбалка' }),

},

{

id: '4',

icon: '',

title: 'Природа',

onPress: () =>

navigation.navigate('Activities', {

tag: 'nature',

title: 'Природа Камчатки',

}),

},

{

id: '5',

icon: '🚁',

title: 'Вертолёты',

onPress: () =>

navigation.navigate('Activities', {

tag: 'helicopter',

title: 'Вертолётные экскурсии',

}),

},

{

id: '6',

icon: '⛵',

title: 'Морские',

onPress: () =>

navigation.navigate('Activities', {

tag: 'sea',

title: 'Морские туры',

}),

},

{

id: '7',

icon: '',

title: 'Джип-Туры',

onPress: () =>

navigation.navigate('Activities', { tag: 'jeep', title: 'Джип-туры' }),

},

{

id: '8',

icon: '',

title: 'Сплавы',

onPress: () =>

navigation.navigate('Activities', {

tag: 'rafting',

title: 'Сплавы и рафтинг',

}),

},

];

const navItems = [

{ id: 'profile', icon: '', label: 'Профиль' },

{ id: 'bulletin', icon: '📢', label: 'Объявления' },

{ id: 'home', icon: '🏠', label: 'Главная' },

{ id: 'favorites', icon: '❤️', label: 'Избранное' },

{ id: 'search', icon: '🔍', label: 'Поиск' },

];

const partnerPhotos = [

{

id: '1',

url: 'https://i.pinimg.com/736x/0a/c2/86/0ac28605a4e075019f0ab47123461bf8.jpg',

title: 'Рыбалка на Камчатке',

source: 'fishingkam.ru',

},

{

id: '2',

url: 'https://i.pinimg.com/originals/8d/e5/ea/8de5ea9c3f38b8601474d54a7379db5b.jpg',

title: 'Туры по Камчатке',

source: 'mestechkokam.ru',

},

{

id: '3',

url: 'https://optim.tildacdn.com/tild6136-3038-4630-a135-633937346462/-/format/webp/noroot.png',

title: 'Северное сияние',

source: 'tildacdn.com',

},

{

id: '4',

url: 'https://media-cdn.tripadvisor.com/media/photo-s/10/15/73/91/caption.jpg',

title: 'Морская рыбалка',

source: 'fishingkam.ru',

},

{

id: '5',

url: 'https://kam-travel.ru/wp-content/uploads/2022/10/na-vertolete-na-kurilskoe-ozero.jpg',

title: 'Вулканы Камчатки',

source: 'mestechkokam.ru',

},

{

id: '6',

url: 'https://i.pinimg.com/originals/69/95/fb/6995fbb8cfbfe78efd7b1fd851d5fdba.jpg',

title: 'Медведи Камчатки',

source: 'wildkamchatka.ru',

},

];

const doublePartnerPhotos = [...partnerPhotos, ...partnerPhotos];

const scrollViewRef = useRef(null);

const currentOffset = useRef(0);

useEffect(() => {

const timer = setInterval(() => {

if (scrollViewRef.current) {

currentOffset.current += 1;

scrollViewRef.current.scrollTo({

x: currentOffset.current,

animated: false,

});

const maxScroll = ((width \* 0.65) / 4 + 8) \* partnerPhotos.length;

if (currentOffset.current >= maxScroll) {

currentOffset.current = 0;

scrollViewRef.current.scrollTo({ x: 0, animated: false });

}

}

}, 30);

return () => clearInterval(timer);

}, []);

const openImage = (image) => {

setSelectedImage(image);

setModalVisible(true);

};

const handleNavPress = (id) => {

setActiveTab(id);

setShowSearchBlock(id === 'search');

if (id === 'profile') {

navigation.navigate('Profile');

} else if (id === 'favorites') {

navigation.navigate('Activities', {

tag: 'favorites',

title: 'Избранное',

});

} else if (id === 'bulletin') {

navigation.navigate('Bulletin');

}

};

const handleSearch = () => {

console.log('Search filters:', searchFilters);

navigation.navigate('Activities', {

tag: 'search',

title: 'Результаты поиска',

filters: searchFilters

});

};

const handleFilterChange = (name, value) => {

setSearchFilters((prev) => ({

...prev,

[name]: value,

}));

};

return (

<ImageBackground

source={{

uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg',

}}

style={styles.background}

resizeMode="cover">

<View style={styles.overlay} />

<ScrollView

contentContainerStyle={styles.container}

keyboardShouldPersistTaps="handled">

<View style={styles.header}>

<Text style={styles.mainTitle}>Здесь начинается Россия</Text>

<Text style={styles.subTitle}>Камчатка - земля огня и льда</Text>

<TouchableOpacity

style={styles.emergencyButton}

onPress={() => navigation.navigate('Emergency')}>

<Text style={styles.emergencyButtonText}>SOS</Text>

</TouchableOpacity>

</View>

{showSearchBlock && (

<View style={styles.tourBlock}>

<View style={styles.searchHeader}>

<Text style={styles.gridTitle}>Найти тур</Text>

<Text style={styles.searchSubtitle}>

Открой для себя дикую природу Камчатки

</Text>

</View>

<View style={styles.searchRow}>

<TextInput

style={styles.searchInput}

placeholder="Куда хотите поехать?"

placeholderTextColor="#a0aec0"

onChangeText={(text) => handleFilterChange('query', text)}

returnKeyType="search"

onSubmitEditing={handleSearch}

/>

<TouchableOpacity

style={styles.searchButton}

onPress={handleSearch}>

<Text style={styles.searchButtonText}>Найти</Text>

</TouchableOpacity>

</View>

<View style={styles.filtersGrid}>

<View style={styles.filterItem}>

<Text style={styles.filterLabel}>Тип тура</Text>

<TextInput

style={styles.filterInput}

placeholder="Поход, экскурсия, рыбалка..."

placeholderTextColor="#a0aec0"

value={searchFilters.type}

onChangeText={(text) => handleFilterChange('type', text)}

/>

</View>

<View style={styles.filterItem}>

<Text style={styles.filterLabel}>Даты</Text>

<TextInput

style={styles.filterInput}

placeholder="Выберите даты"

placeholderTextColor="#a0aec0"

value={searchFilters.dates}

onChangeText={(text) => handleFilterChange('dates', text)}

/>

</View>

<View style={styles.filterItem}>

<Text style={styles.filterLabel}>Количество людей</Text>

<TextInput

style={styles.filterInput}

placeholder="Сколько человек?"

placeholderTextColor="#a0aec0"

value={searchFilters.people}

onChangeText={(text) => handleFilterChange('people', text)}

keyboardType="numeric"

/>

</View>

<View style={styles.filterItem}>

<Text style={styles.filterLabel}>Бюджет</Text>

<TextInput

style={styles.filterInput}

placeholder="Максимальная сумма"

placeholderTextColor="#a0aec0"

value={searchFilters.price}

onChangeText={(text) => handleFilterChange('price', text)}

keyboardType="numeric"

/>

</View>

</View>

<TouchableOpacity style={styles.applyButton} onPress={handleSearch}>

<Text style={styles.applyButtonText}>Найти подходящий тур</Text>

</TouchableOpacity>

</View>

)}

{!showSearchBlock && (

<View style={styles.activitiesContainer}>

<Text style={styles.sectionTitle}>Активности Камчатки</Text>

<View style={styles.activitiesGrid}>

{activities.map((activity) => (

<TouchableOpacity

key={activity.id}

style={[

styles.activityCard,

activeTab === activity.id && styles.activeActivityCard,

]}

onPress={activity.onPress}>

<Text style={[styles.activityIcon, styles.whiteIcon]}>

{activity.icon}

</Text>

<Text style={styles.activityTitle}>{activity.title}</Text>

</TouchableOpacity>

))}

</View>

</View>

)}

<View style={styles.carouselContainer}>

<Text style={styles.sectionTitle}>

Камчатка глазами путешественников

</Text>

<View style={styles.carouselWrapper}>

<ScrollView

ref={scrollViewRef}

horizontal

showsHorizontalScrollIndicator={false}

contentContainerStyle={styles.carousel}

scrollEventThrottle={16}>

{doublePartnerPhotos.map((photo, index) => (

<TouchableOpacity

key={`${photo.id}-${index}`}

onPress={() => openImage(photo)}

activeOpacity={0.8}>

<View style={styles.carouselCard}>

<Image

source={{ uri: photo.url }}

style={styles.carouselItem}

resizeMode="cover"

/>

</View>

</TouchableOpacity>

))}

</ScrollView>

</View>

</View>

</ScrollView>

<View

style={[

styles.navBar,

{

bottom: keyboardHeight > 0 ? keyboardHeight + 10 : 40,

},

]}>

{navItems.map((item) => (

<TouchableOpacity

key={item.id}

style={[

styles.navButton,

activeTab === item.id && styles.activeNavButton,

]}

onPress={() => handleNavPress(item.id)}

activeOpacity={0.7}>

<Text

style={[

styles.navIcon,

activeTab === item.id

? styles.activeNavIcon

: styles.inactiveNavIcon,

]}>

{item.icon}

</Text>

</TouchableOpacity>

))}

</View>

<Modal

animationType="fade"

transparent={true}

visible={modalVisible}

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

<View style={styles.modalContainer}>

{selectedImage && (

<>

<Image

source={{ uri: selectedImage.url }}

style={styles.fullImage}

resizeMode="contain"

/>

<View style={styles.imageInfo}>

<Text style={styles.modalTitle}>{selectedImage.title}</Text>

<Text style={styles.modalSource}>

Источник: {selectedImage.source}

</Text>

</View>

<TouchableOpacity

style={styles.closeButton}

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

<Text style={styles.closeButtonText}>✕</Text>

</TouchableOpacity>

</>

)}

</View>

</Modal>

</ImageBackground>

);

};

const ActivitiesScreen = ({ route, navigation }) => {

const { tag, title, filters } = route.params;

const [tourModalVisible, setTourModalVisible] = useState(false);

const [selectedTour, setSelectedTour] = useState(null);

const [selectedDate, setSelectedDate] = useState('2023-07-20');

const [peopleCount, setPeopleCount] = useState(2);

const [activeTab, setActiveTab] = useState('description');

const [tours, setTours] = useState(ALL\_ACTIVITIES);

const [filterModalVisible, setFilterModalVisible] = useState(false);

const [sortOption, setSortOption] = useState('popularity');

const [difficultyFilter, setDifficultyFilter] = useState('all');

const [priceRange, setPriceRange] = useState([0, 50000]);

// Фильтрация туров по тегу

const filteredTours =

tag === 'favorites'

? tours.filter((tour) => tour.isFavorite)

: tours.filter((tour) => tour.tag === tag);

const applyFilters = () => {

let result = [...ALL\_ACTIVITIES];

if (tag === 'favorites') {

result = result.filter(tour => tour.isFavorite);

} else if (tag !== 'all') {

result = result.filter(tour => tour.tag === tag);

}

if (filters?.type) {

result = result.filter(tour =>

tour.title.toLowerCase().includes(filters.type.toLowerCase()) ||

tour.description.toLowerCase().includes(filters.type.toLowerCase())

);

}

if (difficultyFilter !== 'all') {

result = result.filter(tour => tour.difficulty === difficultyFilter);

}

result = result.filter(tour =>

tour.price >= priceRange[0] && tour.price <= priceRange[1]

);

// Сортировка

if (sortOption === 'price-asc') {

result.sort((a, b) => a.price - b.price);

} else if (sortOption === 'price-desc') {

result.sort((a, b) => b.price - a.price);

} else if (sortOption === 'rating') {

result.sort((a, b) => b.rating - a.rating);

}

return result;

};

const finalTours = applyFilters();

const openTourModal = (tour) => {

setSelectedTour(tour);

setTourModalVisible(true);

};

const calculateTotalPrice = () => {

if (!selectedTour) return 0;

return selectedTour.price \* peopleCount;

};

const toggleFavorite = (id) => {

setTours((prev) =>

prev.map((tour) =>

tour.id === id ? { ...tour, isFavorite: !tour.isFavorite } : tour

)

);

};

const renderTourCard = ({ item }) => (

<TouchableOpacity

style={styles.listTourCard}

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

<View style={styles.listImageContainer}>

<Image source={{ uri: item.image }} style={styles.listTourImage} />

{item.spotsLeft > 0 && item.spotsLeft < 5 && (

<View style={styles.spotsLeftBadge}>

<Text style={styles.spotsLeftText}>

Осталось {item.spotsLeft} места

</Text>

</View>

)}

<TouchableOpacity

style={styles.favoriteButton}

onPress={() => toggleFavorite(item.id)}>

<Text

style={[

styles.favoriteIcon,

item.isFavorite && styles.favoriteActive,

]}>

{item.isFavorite ? '❤️' : '🤍'}

</Text>

</TouchableOpacity>

</View>

<View style={styles.listTourInfo}>

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

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

<View style={styles.includedContainer}>

{item.included.slice(0, 3).map((service, index) => (

<Text key={index} style={styles.includedService}>

{service}

</Text>

))}

</View>

<View style={styles.listPriceRatingContainer}>

<View>

<Text style={styles.listTourPrice}>

{item.price.toLocaleString('ru-RU')} ₽

</Text>

<Text style={styles.pricePerPerson}>за человека</Text>

</View>

<View style={styles.listRatingContainer}>

<Text style={styles.listRating}>★ {item.rating}</Text>

<Text style={styles.listReviews}>({item.reviews})</Text>

</View>

</View>

<View style={styles.bookingFooter}>

<Text style={styles.cancellationInfo}>✅ {item.cancellation}</Text>

<TouchableOpacity

style={styles.bookNowButton}

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

<Text style={styles.bookNowText}>Выбрать даты</Text>

</TouchableOpacity>

</View>

</View>

</TouchableOpacity>

);

const renderModalContent = () => {

if (!selectedTour) return null;

return (

<View style={styles.tourModalContent}>

<ScrollView

horizontal

pagingEnabled

showsHorizontalScrollIndicator={false}

style={styles.tourGallery}>

{[selectedTour.image].map((img, index) => (

<Image

key={index}

source={{ uri: img }}

style={styles.tourGalleryImage}

resizeMode="cover"

/>

))}

</ScrollView>

<View style={styles.tourModalInfo}>

<View style={styles.tourHeader}>

<Text style={styles.tourModalTitle}>{selectedTour.title}</Text>

<TouchableOpacity

style={styles.modalFavoriteButton}

onPress={() => toggleFavorite(selectedTour.id)}>

<Text

style={[

styles.modalFavoriteIcon,

selectedTour.isFavorite && styles.favoriteActive,

]}>

{selectedTour.isFavorite ? '❤️' : '🤍'}

</Text>

</TouchableOpacity>

</View>

<Text style={styles.tourModalLocation}>

📍 {selectedTour.location}

</Text>

<View style={styles.tourMetaContainer}>

<View style={styles.ratingContainer}>

<Text style={styles.tourModalRating}>

★ {selectedTour.rating}

</Text>

<Text style={styles.tourModalReviews}>

({selectedTour.reviews} отзывов)

</Text>

</View>

<View

style={[

styles.difficultyBadge,

selectedTour.difficulty === 'Легкая' && styles.difficultyEasy,

selectedTour.difficulty === 'Средняя' &&

styles.difficultyMedium,

selectedTour.difficulty === 'Сложная' && styles.difficultyHard,

]}>

<Text style={styles.difficultyText}>

{selectedTour.difficulty}

</Text>

</View>

</View>

<View style={styles.tabContainer}>

<TouchableOpacity

style={[

styles.tabButton,

activeTab === 'description' && styles.activeTab,

]}

onPress={() => setActiveTab('description')}>

<Text style={styles.tabText}>Описание</Text>

</TouchableOpacity>

<TouchableOpacity

style={[

styles.tabButton,

activeTab === 'details' && styles.activeTab,

]}

onPress={() => setActiveTab('details')}>

<Text style={styles.tabText}>Детали</Text>

</TouchableOpacity>

<TouchableOpacity

style={[

styles.tabButton,

activeTab === 'reviews' && styles.activeTab,

]}

onPress={() => setActiveTab('reviews')}>

<Text style={styles.tabText}>Отзывы</Text>

</TouchableOpacity>

<TouchableOpacity

style={[

styles.tabButton,

activeTab === 'map' && styles.activeTab,

]}

onPress={() => setActiveTab('map')}>

<Text style={styles.tabText}>Карта</Text>

</TouchableOpacity>

</View>

{activeTab === 'description' && (

<>

<Text style={styles.sectionHeader}>Что включено:</Text>

<View style={styles.includedGrid}>

{selectedTour.included.map((item, index) => (

<View key={index} style={styles.includedItem}>

<Text style={styles.includedIcon}>

{item.split(' ')[0]}

</Text>

<Text style={styles.includedText}>

{item.split(' ').slice(1).join(' ')}

</Text>

</View>

))}

</View>

<Text style={styles.sectionHeader}>Описание:</Text>

<ScrollView style={styles.descriptionScroll}>

<Text style={styles.tourModalDescription}>

{selectedTour.description}

</Text>

</ScrollView>

</>

)}

{activeTab === 'details' && (

<View style={styles.tourDetailsGrid}>

<View style={styles.detailItem}>

<Icon name="access-time" size={20} color="#4a5568" />

<Text style={styles.detailText}>{selectedTour.duration}</Text>

</View>

<View style={styles.detailItem}>

<Icon name="people" size={20} color="#4a5568" />

<Text style={styles.detailText}>До 8 человек</Text>

</View>

<View style={styles.detailItem}>

<Icon name="cancel" size={20} color="#4a5568" />

<Text style={styles.detailText}>

{selectedTour.cancellation}

</Text>

</View>

<View style={styles.detailItem}>

<Icon name="landscape" size={20} color="#4a5568" />

<Text style={styles.detailText}>Уникальные пейзажи</Text>

</View>

</View>

)}

{activeTab === 'reviews' && (

<View style={styles.reviewsContainer}>

<Text style={styles.sectionHeader}>Отзывы путешественников:</Text>

<View style={styles.reviewItem}>

<View style={styles.reviewHeader}>

<Text style={styles.reviewAuthor}>Алексей Петров</Text>

<Text style={styles.reviewDate}>15 июля 2023</Text>

</View>

<View style={styles.reviewRating}>

<Text style={styles.reviewStars}>★★★★★</Text>

<Text style={styles.reviewRatingValue}>5.0</Text>

</View>

<Text style={styles.reviewText}>

Невероятное восхождение! Виды с вершины просто захватывают

дух. Гид Антон - профессионал своего дела, все объяснил и

показал. Обязательно вернусь сюда снова!

</Text>

</View>

<View style={styles.reviewItem}>

<View style={styles.reviewHeader}>

<Text style={styles.reviewAuthor}>Мария Иванова</Text>

<Text style={styles.reviewDate}>10 июля 2023</Text>

</View>

<View style={styles.reviewRating}>

<Text style={styles.reviewStars}>★★★★☆</Text>

<Text style={styles.reviewRatingValue}>4.5</Text>

</View>

<Text style={styles.reviewText}>

Отличный тур, но стоило бы улучшить организацию питания.

В остальном всё прекрасно - виды, гид, программа.

Рекомендую всем любителям природы.

</Text>

</View>

</View>

)}

{activeTab === 'map' && (

<View style={styles.mapContainer}>

<MapView

style={styles.map}

initialRegion={{

latitude: selectedTour.coordinates.latitude,

longitude: selectedTour.coordinates.longitude,

latitudeDelta: 0.5,

longitudeDelta: 0.5,

}}>

<Marker

coordinate={selectedTour.coordinates}

title={selectedTour.title}

description={selectedTour.location}

/>

</MapView>

<Text style={styles.mapNote}>

Точка старта: {selectedTour.location}

</Text>

</View>

)}

<View style={styles.peopleSelector}>

<Text style={styles.selectorLabel}>Количество людей:</Text>

<View style={styles.selectorButtons}>

<TouchableOpacity

style={styles.selectorButton}

onPress={() => setPeopleCount(Math.max(1, peopleCount - 1))}>

<Text style={styles.selectorButtonText}>-</Text>

</TouchableOpacity>

<Text style={styles.peopleCount}>{peopleCount}</Text>

<TouchableOpacity

style={styles.selectorButton}

onPress={() => setPeopleCount(peopleCount + 1)}>

<Text style={styles.selectorButtonText}>+</Text>

</TouchableOpacity>

</View>

</View>

<View style={styles.totalPriceContainer}>

<Text style={styles.totalPriceLabel}>Итого:</Text>

<Text style={styles.totalPriceValue}>

{calculateTotalPrice().toLocaleString('ru-RU')} ₽

</Text>

</View>

<TouchableOpacity

style={styles.tourModalBookButton}

onPress={() =>

console.log('Бронирование тура:', selectedTour.title)

}>

<Text style={styles.tourModalBookButtonText}>

Забронировать сейчас

</Text>

</TouchableOpacity>

</View>

<TouchableOpacity

style={styles.tourCloseButton}

onPress={() => setTourModalVisible(false)}>

<Text style={styles.tourCloseButtonText}>✕</Text>

</TouchableOpacity>

</View>

);

};

const renderFilterModal = () => (

<View style={styles.filterModal}>

<Text style={styles.filterModalTitle}>Фильтры</Text>

<View style={styles.filterSection}>

<Text style={styles.filterLabel}>Сложность</Text>

<View style={styles.filterOptions}>

{['all', 'Легкая', 'Средняя', 'Сложная'].map(level => (

<TouchableOpacity

key={level}

style={[

styles.filterOption,

difficultyFilter === level && styles.selectedFilterOption

]}

onPress={() => setDifficultyFilter(level)}>

<Text style={styles.filterOptionText}>

{level === 'all' ? 'Все' : level}

</Text>

</TouchableOpacity>

))}

</View>

</View>

<View style={styles.filterSection}>

<Text style={styles.filterLabel}>Цена, руб</Text>

<View style={styles.priceRangeContainer}>

<Text style={styles.priceRangeLabel}>{priceRange[0].toLocaleString('ru-RU')}</Text>

<View style={styles.sliderContainer}>

<View style={[styles.sliderTrack, { width: `${(priceRange[1] / 50000) \* 100}%` }]} />

</View>

<Text style={styles.priceRangeLabel}>{priceRange[1].toLocaleString('ru-RU')}</Text>

</View>

</View>

<View style={styles.filterButtons}>

<TouchableOpacity

style={styles.filterResetButton}

onPress={() => {

setDifficultyFilter('all');

setPriceRange([0, 50000]);

}}>

<Text style={styles.filterResetText}>Сбросить</Text>

</TouchableOpacity>

<TouchableOpacity

style={styles.filterApplyButton}

onPress={() => setFilterModalVisible(false)}>

<Text style={styles.filterApplyText}>Применить</Text>

</TouchableOpacity>

</View>

</View>

);

return (

<ImageBackground

source={{

uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg',

}}

style={styles.background}

resizeMode="cover">

<View style={styles.overlay} />

<ScrollView contentContainerStyle={styles.listContainer}>

<View style={styles.listHeader}>

<TouchableOpacity

onPress={() => navigation.goBack()}

style={styles.backButton}>

<Text style={styles.backButtonText}>←</Text>

</TouchableOpacity>

<Text style={styles.listMainTitle}>{title}</Text>

</View>

<View style={styles.listFilterContainer}>

<TouchableOpacity

style={styles.listFilterButton}

onPress={() => setFilterModalVisible(true)}>

<Icon name="filter-list" size={20} color="#FFF" />

<Text style={styles.listFilterButtonText}>Фильтры</Text>

</TouchableOpacity>

<TouchableOpacity style={styles.listSortButton}>

<Text style={styles.listSortButtonText}>По популярности ▼</Text>

</TouchableOpacity>

</View>

{finalTours.length > 0 ? (

<FlatList

data={finalTours}

renderItem={renderTourCard}

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

scrollEnabled={false}

contentContainerStyle={styles.listContent}

/>

) : (

<View style={styles.noResultsContainer}>

<Icon name="search-off" size={50} color="#FFF" />

<Text style={styles.noResultsText}>

По вашему запросу туров не найдено

</Text>

<TouchableOpacity

style={styles.resetFiltersButton}

onPress={() => {

setDifficultyFilter('all');

setPriceRange([0, 50000]);

setFilterModalVisible(false);

}}>

<Text style={styles.resetFiltersText}>Сбросить фильтры</Text>

</TouchableOpacity>

</View>

)}

</ScrollView>

<Modal

animationType="fade"

transparent={true}

visible={tourModalVisible}

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

<View style={styles.tourModalContainer}>{renderModalContent()}</View>

</Modal>

<Modal

animationType="slide"

transparent={true}

visible={filterModalVisible}

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

<View style={styles.filterModalContainer}>

{renderFilterModal()}

</View>

</Modal>

</ImageBackground>

);

};

const BulletinBoardScreen = ({ navigation }) => {

const [ads, setAds] = useState([]);

const [filteredAds, setFilteredAds] = useState([]);

const [searchQuery, setSearchQuery] = useState('');

const [activeCategory, setActiveCategory] = useState('all');

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

const [addModalVisible, setAddModalVisible] = useState(false);

const [newAd, setNewAd] = useState({

title: '',

description: '',

category: 'tours',

price: '',

contact: '',

});

useEffect(() => {

setLoading(true);

setTimeout(() => {

setAds(MOCK\_ADS);

setFilteredAds(MOCK\_ADS);

setLoading(false);

}, 500);

}, []);

useEffect(() => {

let result = ads;

if (activeCategory !== 'all') {

result = result.filter((ad) => ad.category === activeCategory);

}

if (searchQuery) {

const query = searchQuery.toLowerCase();

result = result.filter(

(ad) =>

ad.title.toLowerCase().includes(query) ||

ad.description.toLowerCase().includes(query)

);

}

setFilteredAds(result);

}, [searchQuery, activeCategory, ads]);

const toggleFavorite = (id) => {

setAds((prev) =>

prev.map((ad) =>

ad.id === id ? { ...ad, isFavorite: !ad.isFavorite } : ad

)

);

};

const handleAddAd = () => {

if (!newAd.title.trim()) {

Alert.alert('Ошибка', 'Заголовок объявления обязателен');

return;

}

const newAdItem = {

id: Date.now().toString(),

title: newAd.title,

description: newAd.description,

category: newAd.category,

date: new Date().toISOString().split('T')[0],

source: 'Мое объявление',

url: '',

isFavorite: false,

price: newAd.price,

contact: newAd.contact,

};

setAds((prev) => [...prev, newAdItem]);

setFilteredAds((prev) => [...prev, newAdItem]);

setAddModalVisible(false);

setNewAd({

title: '',

description: '',

category: 'tours',

price: '',

contact: '',

});

Alert.alert('Успех', 'Объявление успешно добавлено');

};

const renderAdItem = ({ item }) => (

<TouchableOpacity

style={styles.adCard}

onPress={() => {

if (item.url) {

Linking.openURL(item.url);

} else {

Alert.alert(

item.title,

`${item.description}\n\nЦена: ${item.price || 'не указана'}\nКонтакт: ${item.contact || 'не указан'}`

);

}

}}>

<View style={styles.adHeader}>

<View style={styles.adCategoryBadge}>

<Icon

name={AD\_CATEGORIES.find(c => c.id === item.category)?.icon}

size={16}

color="#FFF"

/>

<Text style={styles.adCategory}>

{AD\_CATEGORIES.find((c) => c.id === item.category)?.name}

</Text>

</View>

<Text style={styles.adDate}>{item.date}</Text>

</View>

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

{item.price && <Text style={styles.adPrice}>Цена: {item.price}</Text>}

<Text style={styles.adDescription}>{item.description}</Text>

<View style={styles.adFooter}>

<Text style={styles.adSource}>Источник: {item.source}</Text>

<TouchableOpacity

onPress={() => toggleFavorite(item.id)}

style={styles.favoriteButton}>

<Text

style={[

styles.favoriteIcon,

item.isFavorite && styles.favoriteActive,

]}>

{item.isFavorite ? '❤️' : '🤍'}

</Text>

</TouchableOpacity>

</View>

</TouchableOpacity>

);

return (

<ImageBackground

source={{

uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg',

}}

style={styles.background}

resizeMode="cover">

<View style={styles.overlay} />

<View style={styles.listHeader}>

<TouchableOpacity

onPress={() => navigation.goBack()}

style={styles.backButton}>

<Text style={styles.backButtonText}>←</Text>

</TouchableOpacity>

<Text style={styles.listMainTitle}>Доска объявлений</Text>

</View>

<View style={styles.searchContainer}>

<TextInput

style={styles.searchInput}

placeholder="Поиск объявлений..."

placeholderTextColor="#718096"

value={searchQuery}

onChangeText={setSearchQuery}

/>

</View>

<ScrollView

horizontal

showsHorizontalScrollIndicator={false}

style={styles.categoriesContainer}

contentContainerStyle={styles.categoriesContent}>

{AD\_CATEGORIES.map((category) => (

<TouchableOpacity

key={category.id}

style={[

styles.categoryButton,

activeCategory === category.id && styles.activeCategory,

]}

onPress={() => setActiveCategory(category.id)}>

<Icon name={category.icon} size={20} color="#FFF" style={styles.categoryIcon} />

<Text style={styles.categoryText}>{category.name}</Text>

</TouchableOpacity>

))}

</ScrollView>

{loading ? (

<View style={styles.loadingContainer}>

<ActivityIndicator size="large" color="#FFF" />

<Text style={styles.loadingText}>Загрузка объявлений...</Text>

</View>

) : filteredAds.length > 0 ? (

<FlatList

data={filteredAds}

renderItem={renderAdItem}

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

contentContainerStyle={styles.adsContainer}

/>

) : (

<View style={styles.noResultsContainer}>

<Icon name="error-outline" size={50} color="#FFF" />

<Text style={styles.noResultsText}>Объявлений не найдено</Text>

</View>

)}

<TouchableOpacity

style={styles.addButton}

onPress={() => setAddModalVisible(true)}>

<Text style={styles.addButtonText}>+</Text>

</TouchableOpacity>

<Modal

animationType="slide"

transparent={true}

visible={addModalVisible}

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

<View style={styles.modalContainer}>

<View style={styles.modalContent}>

<Text style={styles.modalTitle}>Добавить объявление</Text>

<TextInput

style={styles.input}

placeholder="Заголовок \*"

placeholderTextColor="#666"

value={newAd.title}

onChangeText={(text) => setNewAd({ ...newAd, title: text })}

/>

<TextInput

style={[styles.input, styles.multilineInput]}

placeholder="Описание"

placeholderTextColor="#666"

value={newAd.description}

onChangeText={(text) => setNewAd({ ...newAd, description: text })}

multiline

/>

<Text style={styles.label}>Категория</Text>

<View style={styles.categoryOptions}>

{AD\_CATEGORIES.filter(cat => cat.id !== 'all').map(category => (

<TouchableOpacity

key={category.id}

style={[

styles.categoryOption,

newAd.category === category.id && styles.selectedCategory,

]}

onPress={() => setNewAd({ ...newAd, category: category.id })}>

<Icon name={category.icon} size={20} color="#2d3748" />

<Text style={styles.categoryOptionText}>{category.name}</Text>

</TouchableOpacity>

))}

</View>

<View style={styles.inputRow}>

<View style={styles.inputColumn}>

<Text style={styles.label}>Цена</Text>

<TextInput

style={styles.input}

placeholder="Укажите цену"

placeholderTextColor="#666"

value={newAd.price}

onChangeText={(text) => setNewAd({ ...newAd, price: text })}

keyboardType="numeric"

/>

</View>

<View style={styles.inputColumn}>

<Text style={styles.label}>Контакт</Text>

<TextInput

style={styles.input}

placeholder="Телефон или @ник"

placeholderTextColor="#666"

value={newAd.contact}

onChangeText={(text) => setNewAd({ ...newAd, contact: text })}

/>

</View>

</View>

<View style={styles.modalButtons}>

<TouchableOpacity

style={[styles.modalButton, styles.cancelButton]}

onPress={() => setAddModalVisible(false)}>

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

</TouchableOpacity>

<TouchableOpacity

style={[styles.modalButton, styles.saveButton]}

onPress={handleAddAd}>

<Text style={styles.saveButtonText}>Добавить</Text>

</TouchableOpacity>

</View>

</View>

</View>

</Modal>

</ImageBackground>

);

};

const ProfileScreen = ({ navigation }) => {

return (

<ImageBackground

source={{

uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg',

}}

style={styles.background}

resizeMode="cover">

<View style={styles.overlay} />

<ScrollView contentContainerStyle={styles.profileContainer}>

<View style={styles.profileHeader}>

<TouchableOpacity

onPress={() => navigation.goBack()}

style={styles.backButton}>

<Text style={styles.backButtonText}>←</Text>

</TouchableOpacity>

<Text style={styles.profileTitle}>Личный кабинет</Text>

</View>

<View style={styles.profileCard}>

<Image

source={{ uri: 'https://randomuser.me/api/portraits/men/1.jpg' }}

style={styles.profileAvatar}

/>

<View style={styles.profileInfo}>

<Text style={styles.profileName}>Иван Петров</Text>

<Text style={styles.profileEmail}>ivan@example.com</Text>

<Text style={styles.profilePhone}>+7 (912) 345-67-89</Text>

</View>

<TouchableOpacity style={styles.editButton}>

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

</TouchableOpacity>

</View>

<View style={styles.profileSection}>

<Text style={styles.sectionTitle}>Мои бронирования</Text>

<View style={styles.bookingCard}>

<Text style={styles.bookingTitle}>

Восхождение на Авачинский вулкан

</Text>

<Text style={styles.bookingDate}>20.07.2023</Text>

<Text style={styles.bookingStatus}>Подтверждено</Text>

</View>

<View style={styles.bookingCard}>

<Text style={styles.bookingTitle}>

Экспедиция к Ключевской сопке

</Text>

<Text style={styles.bookingDate}>25.08.2023</Text>

<Text style={styles.bookingStatus}>Ожидает оплаты</Text>

</View>

</View>

<View style={styles.profileSection}>

<Text style={styles.sectionTitle}>Мои инструменты</Text>

<TouchableOpacity

style={styles.toolCard}

onPress={() => navigation.navigate('OfflineMaps')}>

<Icon name="map" size={24} color="#3182ce" />

<Text style={styles.toolTitle}>Офлайн-карты</Text>

<Text style={styles.toolDescription}>Скачайте карты для использования без интернета</Text>

</TouchableOpacity>

<TouchableOpacity

style={styles.toolCard}

onPress={() => navigation.navigate('Emergency')}>

<Icon name="warning" size={24} color="#e53e3e" />

<Text style={styles.toolTitle}>Экстренные контакты</Text>

<Text style={styles.toolDescription}>Важные телефоны и службы Камчатки</Text>

</TouchableOpacity>

</View>

<View style={styles.profileSection}>

<Text style={styles.sectionTitle}>Избранное</Text>

<TouchableOpacity

style={styles.favoriteCard}

onPress={() =>

navigation.navigate('Activities', {

tag: 'favorites',

title: 'Избранное',

})

}>

<Icon name="favorite" size={24} color="#e53e3e" />

<Text style={styles.favoriteTitle}>Избранные туры</Text>

</TouchableOpacity>

<TouchableOpacity

style={styles.favoriteCard}

onPress={() => navigation.navigate('Bulletin')}>

<Icon name="bookmark" size={24} color="#3182ce" />

<Text style={styles.favoriteTitle}>Избранные объявления</Text>

</TouchableOpacity>

</View>

</ScrollView>

<View style={styles.profileNavBar}>

<TouchableOpacity

style={styles.navButton}

onPress={() => navigation.navigate('Home')}>

<Text style={styles.navIcon}>🏠</Text>

</TouchableOpacity>

<TouchableOpacity

style={[styles.navButton, styles.activeNavButton]}

onPress={() => navigation.navigate('Profile')}>

<Text style={[styles.navIcon, styles.activeNavIcon]}>👤</Text>

</TouchableOpacity>

</View>

</ImageBackground>

);

};

const EmergencyScreen = ({ navigation }) => {

const handleCall = (phone) => {

Alert.alert(

'Экстренный вызов',

`Вы действительно хотите позвонить по номеру ${phone}?`,

[

{ text: 'Отмена', style: 'cancel' },

{ text: 'Позвонить', onPress: () => Linking.openURL(`tel:${phone}`) }

]

);

};

return (

<ImageBackground

source={{

uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg',

}}

style={styles.background}

resizeMode="cover">

<View style={styles.overlay} />

<ScrollView contentContainerStyle={styles.emergencyContainer}>

<View style={styles.listHeader}>

<TouchableOpacity

onPress={() => navigation.goBack()}

style={styles.backButton}>

<Text style={styles.backButtonText}>←</Text>

</TouchableOpacity>

<Text style={styles.listMainTitle}>Экстренные контакты</Text>

</View>

<Text style={styles.emergencySubtitle}>

Важные телефоны и службы Камчатки

</Text>

<View style={styles.emergencyList}>

{EMERGENCY\_CONTACTS.map(contact => (

<TouchableOpacity

key={contact.id}

style={styles.emergencyCard}

onPress={() => handleCall(contact.phone)}>

<View style={styles.emergencyIcon}>

<Icon name={contact.icon} size={30} color="#FFF" />

</View>

<View style={styles.emergencyInfo}>

<Text style={styles.emergencyName}>{contact.name}</Text>

<Text style={styles.emergencyPhone}>{contact.phone}</Text>

</View>

<Icon name="call" size={30} color="#3182ce" />

</TouchableOpacity>

))}

</View>

<View style={styles.tipsContainer}>

<Text style={styles.tipsTitle}>Важные советы:</Text>

<View style={styles.tipItem}>

<Icon name="check-circle" size={20} color="#38a169" />

<Text style={styles.tipText}>Сообщайте о своих планах спасателям</Text>

</View>

<View style={styles.tipItem}>

<Icon name="check-circle" size={20} color="#38a169" />

<Text style={styles.tipText}>Берите с собой спутниковый телефон</Text>

</View>

<View style={styles.tipItem}>

<Icon name="check-circle" size={20} color="#38a169" />

<Text style={styles.tipText}>Избегайте встреч с медведями</Text>

</View>

<View style={styles.tipItem}>

<Icon name="check-circle" size={20} color="#38a169" />

<Text style={styles.tipText}>Следите за прогнозом погоды</Text>

</View>

</View>

</ScrollView>

</ImageBackground>

);

};

const OfflineMapsScreen = ({ navigation }) => {

const [downloading, setDownloading] = useState(false);

const [progress, setProgress] = useState(0);

const handleDownload = () => {

setDownloading(true);

// Симуляция загрузки

const interval = setInterval(() => {

setProgress(prev => {

if (prev >= 100) {

clearInterval(interval);

setDownloading(false);

Alert.alert('Успех', 'Карты успешно скачаны!');

return 100;

}

return prev + 10;

});

}, 300);

};

return (

<ImageBackground

source={{

uri: 'https://cs8.pikabu.ru/post\_img/big/2017/05/12/4/1494568737139342763.jpg',

}}

style={styles.background}

resizeMode="cover">

<View style={styles.overlay} />

<ScrollView contentContainerStyle={styles.mapsContainer}>

<View style={styles.listHeader}>

<TouchableOpacity

onPress={() => navigation.goBack()}

style={styles.backButton}>

<Text style={styles.backButtonText}>←</Text>

</TouchableOpacity>

<Text style={styles.listMainTitle}>Офлайн-карты</Text>

</View>

<Text style={styles.mapsDescription}>

Скачайте карты для использования без интернета. Размер пакета: 350 МБ

</Text>

<View style={styles.mapCard}>

<Image

source={{ uri: 'https://i.pinimg.com/originals/8d/e5/ea/8de5ea9c3f38b8601474d54a7379db5b.jpg' }}

style={styles.mapImage}

/>

<View style={styles.mapInfo}>

<Text style={styles.mapTitle}>Карта Камчатского края</Text>

<Text style={styles.mapDetails}>Туристические маршруты, заповедники, вулканы</Text>

<Text style={styles.mapDetails}>Обновлено: 15.07.2023</Text>

</View>

</View>

<View style={styles.mapCard}>

<Image

source={{ uri: 'https://i.pinimg.com/originals/69/95/fb/6995fbb8cfbfe78efd7b1fd851d5fdba.jpg' }}

style={styles.mapImage}

/>

<View style={styles.mapInfo}>

<Text style={styles.mapTitle}>Карта Петропавловска-Камчатского</Text>

<Text style={styles.mapDetails}>Отели, рестораны, транспорт, достопримечательности</Text>

<Text style={styles.mapDetails}>Обновлено: 10.07.2023</Text>

</View>

</View>

{downloading ? (

<View style={styles.downloadContainer}>

<ActivityIndicator size="large" color="#3182ce" />

<Text style={styles.downloadText}>Загрузка карт: {progress}%</Text>

</View>

) : (

<TouchableOpacity

style={styles.downloadButton}

onPress={handleDownload}>

<Icon name="cloud-download" size={24} color="#FFF" />

<Text style={styles.downloadButtonText}>Скачать все карты</Text>

</TouchableOpacity>

)}

</ScrollView>

</ImageBackground>

);

};

const App = () => {

return (

<NavigationContainer>

<Stack.Navigator screenOptions={{ headerShown: false }}>

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

<Stack.Screen name="Activities" component={ActivitiesScreen} />

<Stack.Screen name="Profile" component={ProfileScreen} />

<Stack.Screen name="Bulletin" component={BulletinBoardScreen} />

<Stack.Screen name="Emergency" component={EmergencyScreen} />

<Stack.Screen name="OfflineMaps" component={OfflineMapsScreen} />

</Stack.Navigator>

</NavigationContainer>

);

};

const styles = StyleSheet.create({

background: {

flex: 1,

width: '100%',

height: '100%',

},

overlay: {

...StyleSheet.absoluteFillObject,

backgroundColor: 'rgba(0, 0, 0, 0.3)',

},

container: {

flexGrow: 1,

padding: 15,

paddingBottom: 100,

},

header: {

marginTop: height \* 0.04,

marginBottom: 20,

alignItems: 'center',

position: 'relative',

},

mainTitle: {

fontSize: 28,

fontWeight: 'bold',

color: '#FFF',

textAlign: 'center',

textShadowColor: 'rgba(0, 0, 0, 0.7)',

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 5,

marginBottom: 8,

},

subTitle: {

fontSize: 16,

color: '#FFF',

textAlign: 'center',

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

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 3,

},

emergencyButton: {

position: 'absolute',

top: 0,

right: 20,

backgroundColor: '#e53e3e',

paddingVertical: 6,

paddingHorizontal: 15,

borderRadius: 20,

flexDirection: 'row',

alignItems: 'center',

},

emergencyButtonText: {

color: '#FFF',

fontWeight: 'bold',

fontSize: 16,

},

activitiesContainer: {

marginVertical: 15,

},

sectionTitle: {

fontSize: 15,

fontWeight: '600',

color: '#FFF',

marginBottom: 15,

textAlign: 'center',

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

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 3,

},

activitiesGrid: {

flexDirection: 'row',

flexWrap: 'wrap',

justifyContent: 'space-between',

paddingHorizontal: 5,

},

activityCard: {

width: '24%',

alignItems: 'center',

marginBottom: 10,

padding: 8,

backgroundColor: 'rgba(255, 255, 255, 0.15)',

borderRadius: 12,

borderWidth: 1,

borderColor: 'rgba(255, 255, 255, 0.2)',

},

activeActivityCard: {

backgroundColor: 'rgba(255, 165, 0, 0.4)',

borderColor: 'rgba(255, 165, 0, 0.7)',

},

activityIcon: {

fontSize: 20,

marginBottom: 2,

},

whiteIcon: {

color: '#FFF',

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

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 2,

},

activityTitle: {

fontSize: 10,

fontWeight: '500',

color: '#FFF',

textAlign: 'center',

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

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 2,

},

carouselContainer: {

marginTop: 15,

},

carouselWrapper: {

height: 50,

},

carousel: {

flexDirection: 'row',

alignItems: 'center',

},

carouselCard: {

position: 'relative',

marginRight: 8,

},

carouselItem: {

width: (width \* 0.65) / 4,

height: 180 / 4,

borderRadius: 4,

backgroundColor: '#ccc',

},

navBar: {

position: 'absolute',

left: '5%',

right: '5%',

backgroundColor: 'rgba(120, 120, 120, 0.4)',

flexDirection: 'row',

justifyContent: 'space-around',

alignItems: 'center',

borderRadius: 22,

padding: 6,

borderWidth: 1,

borderColor: 'rgba(255, 255, 255, 0.2)',

shadowColor: '#000',

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

shadowOpacity: 0.2,

shadowRadius: 4,

elevation: 3,

height: 40,

},

navButton: {

width: 32,

height: 32,

borderRadius: 16,

justifyContent: 'center',

alignItems: 'center',

backgroundColor: 'transparent',

},

activeNavButton: {

backgroundColor: 'rgba(255, 255, 255, 0.9)',

},

navIcon: {

fontSize: 18,

textAlign: 'center',

},

activeNavIcon: {

color: '#000',

},

inactiveNavIcon: {

color: 'rgba(255, 255, 255, 0.8)',

},

tourBlock: {

backgroundColor: 'rgba(26, 32, 44, 0.85)',

borderRadius: 16,

padding: 20,

marginVertical: 15,

borderWidth: 2,

borderColor: '#e2b93b',

shadowColor: '#000',

shadowOffset: { width: 0, height: 4 },

shadowOpacity: 0.3,

shadowRadius: 8,

elevation: 8,

},

searchHeader: {

marginBottom: 20,

alignItems: 'center',

},

gridTitle: {

fontSize: 24,

fontWeight: 'bold',

color: '#e2b93b',

marginBottom: 8,

textAlign: 'center',

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

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 2,

},

searchSubtitle: {

fontSize: 14,

color: '#a0aec0',

textAlign: 'center',

},

searchRow: {

flexDirection: 'row',

marginBottom: 20,

},

searchInput: {

flex: 1,

height: 50,

backgroundColor: 'rgba(255, 255, 255, 0.1)',

borderRadius: 12,

borderWidth: 1,

borderColor: '#4a5568',

paddingHorizontal: 20,

color: '#fff',

fontSize: 16,

fontWeight: '500',

},

searchButton: {

width: 100,

height: 50,

backgroundColor: '#e2b93b',

borderRadius: 12,

marginLeft: 10,

justifyContent: 'center',

alignItems: 'center',

shadowColor: '#e2b93b',

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

shadowOpacity: 0.4,

shadowRadius: 4,

elevation: 4,

},

searchButtonText: {

color: '#1a202c',

fontSize: 16,

fontWeight: 'bold',

},

filtersGrid: {

flexDirection: 'row',

flexWrap: 'wrap',

justifyContent: 'space-between',

marginBottom: 20,

},

filterItem: {

width: '48%',

marginBottom: 15,

},

filterLabel: {

color: '#e2b93b',

fontSize: 14,

fontWeight: '600',

marginBottom: 8,

},

filterInput: {

height: 50,

backgroundColor: 'rgba(255, 255, 255, 0.1)',

borderRadius: 12,

borderWidth: 1,

borderColor: '#4a5568',

paddingHorizontal: 15,

color: '#fff',

fontSize: 16,

},

applyButton: {

height: 50,

backgroundColor: '#3182ce',

borderRadius: 12,

justifyContent: 'center',

alignItems: 'center',

shadowColor: '#3182ce',

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

shadowOpacity: 0.4,

shadowRadius: 4,

elevation: 4,

},

applyButtonText: {

color: '#fff',

fontSize: 16,

fontWeight: 'bold',

},

modalContainer: {

flex: 1,

backgroundColor: 'rgba(0, 0, 0, 0.9)',

justifyContent: 'center',

alignItems: 'center',

},

fullImage: {

width: width,

height: height \* 0.7,

},

imageInfo: {

position: 'absolute',

bottom: 40,

left: 20,

right: 20,

backgroundColor: 'rgba(0, 0, 0, 0.6)',

padding: 15,

borderRadius: 10,

},

modalTitle: {

color: '#FFF',

fontSize: 20,

fontWeight: 'bold',

marginBottom: 5,

},

modalSource: {

color: 'rgba(255, 255, 255, 0.7)',

fontSize: 16,

},

closeButton: {

position: 'absolute',

top: 40,

right: 20,

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

width: 40,

height: 40,

borderRadius: 20,

justifyContent: 'center',

alignItems: 'center',

borderWidth: 1,

borderColor: 'rgba(255, 255, 255, 0.3)',

},

closeButtonText: {

color: '#FFF',

fontSize: 24,

fontWeight: 'bold',

},

listContainer: {

flexGrow: 1,

paddingBottom: 20,

},

listHeader: {

flexDirection: 'row',

alignItems: 'center',

justifyContent: 'center',

paddingTop: height \* 0.04,

paddingBottom: 15,

position: 'relative',

},

backButton: {

position: 'absolute',

left: 15,

top: height \* 0.04,

zIndex: 10,

padding: 10,

},

backButtonText: {

fontSize: 28,

color: '#FFF',

fontWeight: 'bold',

},

listMainTitle: {

fontSize: 24,

fontWeight: 'bold',

color: '#FFF',

textAlign: 'center',

textShadowColor: 'rgba(0, 0, 0, 0.7)',

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 5,

},

listFilterContainer: {

flexDirection: 'row',

justifyContent: 'space-between',

paddingHorizontal: 15,

marginBottom: 15,

},

listFilterButton: {

paddingVertical: 8,

paddingHorizontal: 15,

backgroundColor: 'rgba(255,255,255,0.15)',

borderRadius: 20,

borderWidth: 1,

borderColor: '#e2b93b',

flexDirection: 'row',

alignItems: 'center',

},

listFilterButtonText: {

color: '#FFF',

fontWeight: '600',

fontSize: 16,

marginLeft: 5,

},

listSortButton: {

paddingVertical: 8,

paddingHorizontal: 15,

},

listSortButtonText: {

color: '#e2b93b',

fontWeight: '600',

fontSize: 16,

},

listContent: {

paddingHorizontal: 10,

},

listTourCard: {

flexDirection: 'row',

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

overflow: 'hidden',

marginBottom: 15,

height: 160,

shadowColor: '#000',

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

shadowOpacity: 0.2,

shadowRadius: 4,

elevation: 3,

},

listImageContainer: {

width: '40%',

position: 'relative',

},

listTourImage: {

width: '100%',

height: '100%',

resizeMode: 'cover',

},

spotsLeftBadge: {

position: 'absolute',

top: 10,

left: 10,

backgroundColor: '#e53e3e',

paddingHorizontal: 8,

paddingVertical: 4,

borderRadius: 10,

},

spotsLeftText: {

color: '#FFF',

fontSize: 12,

fontWeight: 'bold',

},

favoriteButton: {

position: 'absolute',

top: 10,

right: 10,

backgroundColor: 'rgba(255,255,255,0.7)',

width: 30,

height: 30,

borderRadius: 15,

justifyContent: 'center',

alignItems: 'center',

},

favoriteIcon: {

fontSize: 16,

},

favoriteActive: {

color: '#e53e3e',

},

listTourInfo: {

flex: 1,

padding: 12,

justifyContent: 'space-between',

},

listTourTitle: {

fontSize: 14,

fontWeight: '600',

color: '#2d3748',

marginBottom: 4,

},

listTourLocation: {

fontSize: 12,

color: '#4a5568',

marginBottom: 8,

},

includedContainer: {

flexDirection: 'row',

flexWrap: 'wrap',

marginBottom: 8,

},

includedService: {

fontSize: 11,

color: '#2d3748',

backgroundColor: '#edf2f7',

paddingHorizontal: 6,

paddingVertical: 2,

borderRadius: 4,

marginRight: 6,

marginBottom: 4,

},

listPriceRatingContainer: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

marginBottom: 4,

},

listTourPrice: {

fontSize: 16,

fontWeight: 'bold',

color: '#e53e3e',

},

pricePerPerson: {

fontSize: 10,

color: '#718096',

},

listRatingContainer: {

flexDirection: 'row',

alignItems: 'center',

},

listRating: {

fontSize: 13,

color: '#f6ad55',

fontWeight: 'bold',

marginRight: 4,

},

listReviews: {

fontSize: 11,

color: '#718096',

},

bookingFooter: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

marginTop: 6,

},

cancellationInfo: {

fontSize: 10,

color: '#38a169',

flex: 1,

},

bookNowButton: {

backgroundColor: '#3182ce',

borderRadius: 6,

paddingVertical: 6,

paddingHorizontal: 12,

},

bookNowText: {

color: '#FFF',

fontWeight: '600',

fontSize: 12,

},

noResultsContainer: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

padding: 40,

},

noResultsText: {

fontSize: 18,

color: '#FFF',

textAlign: 'center',

marginTop: 20,

},

resetFiltersButton: {

marginTop: 20,

padding: 10,

backgroundColor: 'rgba(255,255,255,0.2)',

borderRadius: 8,

},

resetFiltersText: {

color: '#FFF',

fontWeight: '600',

},

tourModalContainer: {

flex: 1,

backgroundColor: 'rgba(0, 0, 0, 0.85)',

justifyContent: 'center',

alignItems: 'center',

padding: 20,

},

tourModalContent: {

width: '100%',

backgroundColor: '#fff',

borderRadius: 15,

overflow: 'hidden',

maxHeight: height \* 0.85,

},

tourGallery: {

height: 250,

},

tourGalleryImage: {

width: width - 40,

height: 250,

},

tourModalInfo: {

padding: 20,

},

tourHeader: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

},

tourModalTitle: {

fontSize: 20,

fontWeight: 'bold',

color: '#2d3748',

flex: 1,

},

modalFavoriteButton: {

width: 40,

height: 40,

borderRadius: 20,

backgroundColor: '#f7fafc',

justifyContent: 'center',

alignItems: 'center',

borderWidth: 1,

borderColor: '#e2e8f0',

},

modalFavoriteIcon: {

fontSize: 20,

},

tourModalLocation: {

fontSize: 14,

color: '#4a5568',

marginBottom: 12,

},

tourMetaContainer: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

marginBottom: 12,

},

ratingContainer: {

flexDirection: 'row',

alignItems: 'center',

},

tourModalRating: {

fontSize: 16,

color: '#f6ad55',

fontWeight: 'bold',

marginRight: 5,

},

tourModalReviews: {

fontSize: 14,

color: '#718096',

},

difficultyBadge: {

paddingHorizontal: 10,

paddingVertical: 4,

borderRadius: 12,

},

difficultyEasy: {

backgroundColor: '#c6f6d5',

},

difficultyMedium: {

backgroundColor: '#fefcbf',

},

difficultyHard: {

backgroundColor: '#fed7d7',

},

difficultyText: {

fontWeight: '600',

fontSize: 12,

},

tabContainer: {

flexDirection: 'row',

borderBottomWidth: 1,

borderBottomColor: '#e2e8f0',

marginBottom: 15,

},

tabButton: {

paddingVertical: 10,

paddingHorizontal: 15,

borderBottomWidth: 2,

borderBottomColor: 'transparent',

marginRight: 10,

},

activeTab: {

borderBottomColor: '#3182ce',

},

tabText: {

fontSize: 16,

fontWeight: '500',

color: '#4a5568',

},

sectionHeader: {

fontSize: 18,

fontWeight: 'bold',

color: '#2d3748',

marginTop: 10,

marginBottom: 12,

},

includedGrid: {

flexDirection: 'row',

flexWrap: 'wrap',

justifyContent: 'space-between',

marginBottom: 15,

},

includedItem: {

width: '48%',

flexDirection: 'row',

alignItems: 'center',

marginBottom: 12,

},

includedIcon: {

fontSize: 18,

marginRight: 8,

},

includedText: {

fontSize: 14,

color: '#4a5568',

flex: 1,

},

tourDetailsGrid: {

flexDirection: 'row',

flexWrap: 'wrap',

justifyContent: 'space-between',

},

detailItem: {

width: '100%',

flexDirection: 'row',

alignItems: 'center',

marginBottom: 10,

paddingHorizontal: 5,

},

detailIcon: {

fontSize: 18,

marginRight: 10,

minWidth: 24,

},

detailText: {

fontSize: 14,

color: '#4a5568',

flex: 1,

},

descriptionScroll: {

maxHeight: 150,

marginBottom: 20,

},

tourModalDescription: {

fontSize: 15,

color: '#4a5568',

lineHeight: 22,

},

mapContainer: {

height: 200,

borderRadius: 10,

overflow: 'hidden',

marginBottom: 20,

},

map: {

flex: 1,

},

mapNote: {

fontSize: 12,

color: '#718096',

textAlign: 'center',

marginTop: 5,

},

peopleSelector: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

marginBottom: 20,

padding: 15,

backgroundColor: '#f7fafc',

borderRadius: 10,

},

selectorLabel: {

fontSize: 16,

color: '#2d3748',

},

selectorButtons: {

flexDirection: 'row',

alignItems: 'center',

},

selectorButton: {

width: 36,

height: 36,

borderRadius: 18,

backgroundColor: '#3182ce',

justifyContent: 'center',

alignItems: 'center',

},

selectorButtonText: {

color: '#FFF',

fontSize: 20,

fontWeight: 'bold',

lineHeight: 24,

},

peopleCount: {

fontSize: 18,

fontWeight: 'bold',

marginHorizontal: 15,

minWidth: 30,

textAlign: 'center',

},

totalPriceContainer: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

paddingVertical: 15,

borderTopWidth: 1,

borderTopColor: '#e2e8f0',

borderBottomWidth: 1,

borderBottomColor: '#e2e8f0',

marginBottom: 20,

},

totalPriceLabel: {

fontSize: 18,

fontWeight: '500',

color: '#2d3748',

},

totalPriceValue: {

fontSize: 22,

fontWeight: 'bold',

color: '#3182ce',

},

tourModalBookButton: {

backgroundColor: '#3182ce',

borderRadius: 8,

paddingVertical: 12,

alignItems: 'center',

},

tourModalBookButtonText: {

color: '#FFF',

fontWeight: 'bold',

fontSize: 16,

},

tourCloseButton: {

position: 'absolute',

top: 15,

right: 15,

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

width: 36,

height: 36,

borderRadius: 18,

justifyContent: 'center',

alignItems: 'center',

},

tourCloseButtonText: {

color: '#FFF',

fontSize: 20,

fontWeight: 'bold',

},

reviewsContainer: {

marginBottom: 20,

},

reviewItem: {

backgroundColor: '#f7fafc',

borderRadius: 10,

padding: 15,

marginBottom: 15,

},

reviewHeader: {

flexDirection: 'row',

justifyContent: 'space-between',

marginBottom: 8,

},

reviewAuthor: {

fontSize: 16,

fontWeight: 'bold',

color: '#2d3748',

},

reviewDate: {

fontSize: 14,

color: '#718096',

},

reviewRating: {

flexDirection: 'row',

alignItems: 'center',

marginBottom: 8,

},

reviewStars: {

color: '#f6ad55',

fontSize: 16,

marginRight: 8,

},

reviewRatingValue: {

fontSize: 16,

fontWeight: 'bold',

color: '#f6ad55',

},

reviewText: {

fontSize: 14,

color: '#4a5568',

lineHeight: 20,

},

filterModalContainer: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

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

},

filterModal: {

width: '90%',

backgroundColor: '#FFF',

borderRadius: 15,

padding: 20,

},

filterModalTitle: {

fontSize: 20,

fontWeight: 'bold',

marginBottom: 20,

textAlign: 'center',

color: '#2d3748',

},

filterSection: {

marginBottom: 20,

},

filterLabel: {

fontSize: 16,

fontWeight: '600',

marginBottom: 10,

color: '#2d3748',

},

filterOptions: {

flexDirection: 'row',

flexWrap: 'wrap',

},

filterOption: {

paddingVertical: 8,

paddingHorizontal: 15,

backgroundColor: '#edf2f7',

borderRadius: 20,

marginRight: 10,

marginBottom: 10,

},

selectedFilterOption: {

backgroundColor: '#3182ce',

},

filterOptionText: {

fontSize: 14,

color: '#2d3748',

},

priceRangeContainer: {

flexDirection: 'row',

alignItems: 'center',

justifyContent: 'space-between',

},

priceRangeLabel: {

fontSize: 14,

color: '#4a5568',

},

sliderContainer: {

flex: 1,

height: 4,

backgroundColor: '#e2e8f0',

borderRadius: 2,

marginHorizontal: 10,

position: 'relative',

},

sliderTrack: {

height: 4,

backgroundColor: '#3182ce',

borderRadius: 2,

},

filterButtons: {

flexDirection: 'row',

justifyContent: 'space-between',

marginTop: 10,

},

filterResetButton: {

flex: 1,

padding: 12,

backgroundColor: '#e2e8f0',

borderRadius: 8,

marginRight: 5,

alignItems: 'center',

},

filterResetText: {

color: '#2d3748',

fontWeight: '600',

},

filterApplyButton: {

flex: 1,

padding: 12,

backgroundColor: '#3182ce',

borderRadius: 8,

marginLeft: 5,

alignItems: 'center',

},

filterApplyText: {

color: '#FFF',

fontWeight: '600',

},

searchContainer: {

paddingHorizontal: 15,

marginBottom: 10,

marginTop: 10,

},

searchInput: {

height: 45,

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

paddingHorizontal: 15,

fontSize: 16,

borderWidth: 1,

borderColor: '#e2e8f0',

},

categoriesContainer: {

maxHeight: 40,

marginBottom: 15,

},

categoriesContent: {

paddingHorizontal: 10,

},

categoryButton: {

paddingHorizontal: 15,

paddingVertical: 8,

backgroundColor: 'rgba(255, 255, 255, 0.15)',

borderRadius: 20,

marginRight: 10,

flexDirection: 'row',

alignItems: 'center',

},

activeCategory: {

backgroundColor: '#e2b93b',

},

categoryIcon: {

marginRight: 5,

},

categoryText: {

color: '#FFF',

fontSize: 14,

},

adsContainer: {

paddingHorizontal: 15,

paddingBottom: 100,

},

adCard: {

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 10,

padding: 15,

marginBottom: 15,

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 3,

elevation: 2,

},

adHeader: {

flexDirection: 'row',

justifyContent: 'space-between',

marginBottom: 8,

},

adCategoryBadge: {

flexDirection: 'row',

alignItems: 'center',

backgroundColor: '#3182ce',

paddingHorizontal: 8,

paddingVertical: 3,

borderRadius: 12,

},

adCategory: {

color: '#FFF',

fontWeight: '600',

fontSize: 12,

marginLeft: 5,

},

adDate: {

color: '#718096',

fontSize: 12,

},

adTitle: {

fontSize: 16,

fontWeight: 'bold',

color: '#2d3748',

marginBottom: 5,

},

adPrice: {

fontSize: 15,

fontWeight: 'bold',

color: '#2d3748',

marginBottom: 5,

},

adDescription: {

fontSize: 14,

color: '#4a5568',

marginBottom: 10,

lineHeight: 20,

},

adFooter: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

},

adSource: {

color: '#718096',

fontSize: 12,

},

loadingContainer: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

padding: 20,

},

loadingText: {

color: '#FFF',

fontSize: 16,

marginTop: 10,

},

profileContainer: {

flexGrow: 1,

paddingBottom: 100,

},

profileHeader: {

flexDirection: 'row',

alignItems: 'center',

paddingTop: height \* 0.04,

paddingBottom: 20,

paddingHorizontal: 20,

},

profileTitle: {

fontSize: 24,

fontWeight: 'bold',

color: '#FFF',

textAlign: 'center',

flex: 1,

textShadowColor: 'rgba(0, 0, 0, 0.7)',

textShadowOffset: { width: 1, height: 1 },

textShadowRadius: 5,

},

profileCard: {

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 16,

padding: 20,

margin: 20,

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

},

profileAvatar: {

width: 60,

height: 60,

borderRadius: 30,

marginRight: 15,

},

profileInfo: {

flex: 1,

},

profileName: {

fontSize: 20,

fontWeight: 'bold',

color: '#2d3748',

marginBottom: 5,

},

profileEmail: {

fontSize: 16,

color: '#4a5568',

marginBottom: 3,

},

profilePhone: {

fontSize: 16,

color: '#4a5568',

},

editButton: {

backgroundColor: '#3182ce',

paddingVertical: 8,

paddingHorizontal: 15,

borderRadius: 8,

},

editButtonText: {

color: '#FFF',

fontWeight: '600',

},

profileSection: {

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 16,

padding: 20,

marginHorizontal: 20,

marginBottom: 20,

},

bookingCard: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

paddingVertical: 15,

borderBottomWidth: 1,

borderBottomColor: '#e2e8f0',

},

bookingTitle: {

fontSize: 16,

fontWeight: '600',

color: '#2d3748',

marginBottom: 5,

flex: 1,

},

bookingDate: {

fontSize: 14,

color: '#718096',

marginRight: 10,

},

bookingStatus: {

paddingVertical: 5,

paddingHorizontal: 10,

borderRadius: 12,

backgroundColor: '#c6f6d5',

},

toolCard: {

flexDirection: 'row',

alignItems: 'center',

paddingVertical: 15,

borderBottomWidth: 1,

borderBottomColor: '#e2e8f0',

},

toolTitle: {

fontSize: 16,

fontWeight: '600',

color: '#2d3748',

marginLeft: 10,

flex: 1,

},

toolDescription: {

fontSize: 12,

color: '#718096',

marginLeft: 10,

},

favoriteCard: {

flexDirection: 'row',

alignItems: 'center',

paddingVertical: 15,

borderBottomWidth: 1,

borderBottomColor: '#e2e8f0',

},

favoriteTitle: {

fontSize: 16,

fontWeight: '600',

color: '#2d3748',

marginLeft: 10,

flex: 1,

},

profileNavBar: {

position: 'absolute',

bottom: 70,

left: '5%',

right: '5%',

backgroundColor: 'rgba(120, 120, 120, 0.4)',

flexDirection: 'row',

justifyContent: 'space-around',

borderRadius: 22,

padding: 6,

borderWidth: 1,

borderColor: 'rgba(255, 255, 255, 0.2)',

},

addButton: {

position: 'absolute',

right: 20,

bottom: 80,

width: 60,

height: 60,

borderRadius: 30,

backgroundColor: '#e2b93b',

justifyContent: 'center',

alignItems: 'center',

shadowColor: '#000',

shadowOffset: { width: 0, height: 4 },

shadowOpacity: 0.3,

shadowRadius: 8,

elevation: 8,

zIndex: 100,

},

addButtonText: {

fontSize: 36,

color: '#FFF',

lineHeight: 38,

},

modalContainer: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

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

},

modalContent: {

width: '90%',

backgroundColor: '#FFF',

borderRadius: 16,

padding: 20,

},

modalTitle: {

fontSize: 22,

fontWeight: 'bold',

textAlign: 'center',

marginBottom: 20,

color: '#2d3748',

},

input: {

height: 50,

borderWidth: 1,

borderColor: '#ddd',

borderRadius: 10,

padding: 15,

marginBottom: 15,

fontSize: 16,

backgroundColor: '#f8f9fa',

},

multilineInput: {

height: 100,

textAlignVertical: 'top',

},

label: {

fontSize: 16,

fontWeight: '600',

marginBottom: 10,

color: '#2d3748',

},

categoryOptions: {

flexDirection: 'row',

flexWrap: 'wrap',

marginBottom: 15,

},

categoryOption: {

paddingVertical: 8,

paddingHorizontal: 15,

borderWidth: 1,

borderColor: '#ddd',

borderRadius: 20,

marginRight: 10,

marginBottom: 10,

backgroundColor: '#f8f9fa',

flexDirection: 'row',

alignItems: 'center',

},

selectedCategory: {

backgroundColor: '#e2b93b',

borderColor: '#e2b93b',

},

categoryOptionText: {

fontSize: 14,

color: '#2d3748',

marginLeft: 5,

},

inputRow: {

flexDirection: 'row',

justifyContent: 'space-between',

},

inputColumn: {

width: '48%',

},

modalButtons: {

flexDirection: 'row',

justifyContent: 'space-between',

marginTop: 10,

},

modalButton: {

flex: 1,

paddingVertical: 14,

borderRadius: 10,

alignItems: 'center',

marginHorizontal: 5,

},

cancelButton: {

backgroundColor: '#e2e8f0',

},

cancelButtonText: {

color: '#4a5568',

fontWeight: 'bold',

fontSize: 16,

},

saveButton: {

backgroundColor: '#e2b93b',

},

saveButtonText: {

color: '#FFF',

fontWeight: 'bold',

fontSize: 16,

},

emergencyContainer: {

flexGrow: 1,

paddingBottom: 30,

},

emergencySubtitle: {

fontSize: 16,

color: '#FFF',

textAlign: 'center',

marginBottom: 20,

paddingHorizontal: 20,

},

emergencyList: {

paddingHorizontal: 15,

},

emergencyCard: {

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

padding: 15,

marginBottom: 15,

flexDirection: 'row',

alignItems: 'center',

},

emergencyIcon: {

width: 50,

height: 50,

borderRadius: 25,

backgroundColor: '#3182ce',

justifyContent: 'center',

alignItems: 'center',

marginRight: 15,

},

emergencyInfo: {

flex: 1,

},

emergencyName: {

fontSize: 16,

fontWeight: '600',

color: '#2d3748',

marginBottom: 5,

},

emergencyPhone: {

fontSize: 18,

fontWeight: 'bold',

color: '#3182ce',

},

tipsContainer: {

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

padding: 20,

margin: 15,

},

tipsTitle: {

fontSize: 18,

fontWeight: 'bold',

color: '#2d3748',

marginBottom: 15,

},

tipItem: {

flexDirection: 'row',

alignItems: 'center',

marginBottom: 10,

},

tipText: {

fontSize: 14,

color: '#4a5568',

marginLeft: 10,

flex: 1,

},

mapsContainer: {

flexGrow: 1,

paddingBottom: 30,

},

mapsDescription: {

fontSize: 16,

color: '#FFF',

textAlign: 'center',

marginBottom: 20,

paddingHorizontal: 20,

},

mapCard: {

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

overflow: 'hidden',

marginBottom: 15,

marginHorizontal: 15,

},

mapImage: {

width: '100%',

height: 150,

},

mapInfo: {

padding: 15,

},

mapTitle: {

fontSize: 16,

fontWeight: 'bold',

color: '#2d3748',

marginBottom: 5,

},

mapDetails: {

fontSize: 14,

color: '#4a5568',

marginBottom: 3,

},

downloadContainer: {

alignItems: 'center',

marginTop: 20,

marginHorizontal: 15,

padding: 20,

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

},

downloadText: {

fontSize: 16,

color: '#2d3748',

marginTop: 15,

},

downloadButton: {

flexDirection: 'row',

alignItems: 'center',

justifyContent: 'center',

backgroundColor: '#3182ce',

borderRadius: 12,

padding: 15,

marginHorizontal: 15,

marginTop: 10,

},

downloadButtonText: {

color: '#FFF',

fontWeight: 'bold',

fontSize: 16,

marginLeft: 10,

},

});

export default App;

Основные улучшения и новые функции:

1. Добавлены новые экраны:
   * Экстренные контакты (SOS)
   * Офлайн-карты
   * Улучшенный экран профиля
2. Улучшен дизайн:
   * Иконки для всех категорий и действий
   * Аватары в профиле
   * Визуальное оформление карточек
   * Индикаторы прогресса загрузки
3. Новые функции:
   * Фильтрация туров по сложности и цене
   * Интерактивная карта в деталях тура
   * Возможность скачивания офлайн-карт
   * Система экстренных вызовов
   * Улучшенная форма добавления объявлений
4. Технические улучшения:
   * Оптимизированная навигация
   * Анимации загрузки
   * Валидация форм
   * Реализована сортировка туров
5. Дополнительный контент:
   * Детальные описания активностей
   * Реальные координаты для карты
   * Расширенные отзывы
   * Советы для туристов

Приложение полностью готово к использованию и соответствует всем требованиям для поиска туристических активностей на Камчатке.

App.js (453:6) React Hook useEffect has a missing dependency: 'partnerPhotos.length'. Either include it or remove the dependency array. (react-hooks/exhaustive-deps) (ESLint) App.js (2496:3) Unused style detected: styles.detailIcon (react-native/no-unused-styles) (ESLint) App.js (2681:3) Duplicate key 'filterLabel'. (no-dupe-keys) (ESLint) App.js (2762:3) Duplicate key 'searchInput'. (no-dupe-keys) (ESLint) App.js (3038:3) Duplicate key 'modalContainer'. (no-dupe-keys) (ESLint) App.js (3050:3) Duplicate key 'modalTitle'. (no-dupe-keys) (ESLint) КАК ИСПРАВИТЬ

Мы получили несколько предупреждений ESLint в коде. Давайте исправим их по очереди:

1. \*\*React Hook useEffect has a missing dependency: 'partnerPhotos.length'\*\*

- В `useEffect` внутри `HomeScreen` мы используем `partnerPhotos.length`, но не включили его в зависимости. Это может привести к устаревшим значениям. Добавим `partnerPhotos.length` в массив зависимостей.

2. \*\*Unused style detected: styles.detailIcon\*\*

- В стилях есть `detailIcon`, который не используется в коде. Удалим его.

3. \*\*Duplicate key 'filterLabel'\*\* (и другие дубликаты ключей)

- В объекте стилей есть несколько дубликатов ключей. Это происходит потому, что мы определяем один и тот же ключ несколько раз. Нужно объединить или переименовать дублирующиеся стили.

Давайте исправим эти проблемы:

### 1. Исправление useEffect в HomeScreen

Заменим:

```javascript

useEffect(() => {

const timer = setInterval(() => {

// ... код

}, 30);

return () => clearInterval(timer);

}, []);

```

на:

```javascript

useEffect(() => {

const timer = setInterval(() => {

// ... код

}, 30);

return () => clearInterval(timer);

}, [partnerPhotos.length]); // Добавляем partnerPhotos.length в зависимости

```

### 2. Удаление неиспользуемого стиля

Удалим `detailIcon` из стилей, так как он нигде не используется.

### 3. Устранение дубликатов ключей в стилях

В коде стилей несколько раз встречаются одинаковые имена. Это может быть вызвано тем, что в разных частях кода мы определили стили с одинаковыми именами. Нужно переименовать дублирующиеся стили, чтобы они были уникальными.

Конкретные дубликаты:

- `filterLabel` (встречается дважды)

- `searchInput` (дважды)

- `modalContainer` (дважды)

- `modalTitle` (дважды)

Поскольку у нас большой объект стилей, мы должны переименовать дубликаты, добавив суффиксы, чтобы сделать их уникальными. Однако, если эти стили используются в разных компонентах и не конфликтуют, то можно оставить как есть? Но ESLint ругается, поэтому нужно исправить.

Но в React Native стили определяются в одном большом объекте, поэтому дубликаты ключей недопустимы. Поэтому переименуем:

- Первый `filterLabel` (в HomeScreen) остается `filterLabel`.

- Второй `filterLabel` (в ActivitiesScreen) переименуем в `filterSectionLabel`.

Аналогично:

- `searchInput` в HomeScreen оставляем, а в BulletinBoardScreen переименуем в `bulletinSearchInput`.

- `modalContainer` в HomeScreen (для галереи) оставляем, а в BulletinBoardScreen (для модального окна добавления объявления) переименуем в `addModalContainer`.

- `modalTitle` в HomeScreen (для галереи) оставляем, а в BulletinBoardScreen переименуем в `addModalTitle`.

Также проверим, чтобы при переименовании мы обновили использование этих стилей в компонентах.

Давайте внесем все исправления в код.

Из-за большого объема кода я не могу привести весь исправленный код, но покажу исправленные фрагменты:

### Исправление 1: useEffect в HomeScreen

```javascript

useEffect(() => {

const timer = setInterval(() => {

if (scrollViewRef.current) {

currentOffset.current += 1;

scrollViewRef.current.scrollTo({

x: currentOffset.current,

animated: false,

});

const maxScroll = ((width \* 0.65) / 4 + 8) \* partnerPhotos.length;

if (currentOffset.current >= maxScroll) {

currentOffset.current = 0;

scrollViewRef.current.scrollTo({ x: 0, animated: false });

}

}

}, 30);

return () => clearInterval(timer);

}, [partnerPhotos.length]); // Добавили зависимость

```

### Исправление 2: Удаление неиспользуемого стиля

В объекте стилей удаляем:

```javascript

detailIcon: {

fontSize: 18,

marginRight: 10,

minWidth: 24,

},

```

### Исправление 3: Переименование дубликатов стилей

В стилях:

Заменим:

```javascript

filterLabel: {

fontSize: 16,

fontWeight: '600',

marginBottom: 10,

color: '#2d3748',

},

```

на (для ActivitiesScreen):

```javascript

filterSectionLabel: {

fontSize: 16,

fontWeight: '600',

marginBottom: 10,

color: '#2d3748',

},

```

И в компоненте ActivitiesScreen в функции `renderFilterModal` заменим использование `styles.filterLabel` на `styles.filterSectionLabel`.

Аналогично для других дубликатов:

\*\*Для searchInput:\*\*

В BulletinBoardScreen переименуем стиль `searchInput` в `bulletinSearchInput` и обновим использование в компоненте.

\*\*Для modalContainer:\*\*

В BulletinBoardScreen переименуем `modalContainer` в `addModalContainer`.

\*\*Для modalTitle:\*\*

В BulletinBoardScreen переименуем `modalTitle` в `addModalTitle`.

Теперь исправим код, чтобы он соответствовал этим изменениям.

Поскольку это очень большой файл, я приведу только измененные части:

### ActivitiesScreen: renderFilterModal

```jsx

<View style={styles.filterModal}>

<Text style={styles.filterModalTitle}>Фильтры</Text>

<View style={styles.filterSection}>

<Text style={styles.filterSectionLabel}>Сложность</Text> {/\* Было filterLabel, теперь filterSectionLabel \*/}

{/\* ... \*/}

</View>

{/\* ... \*/}

</View>

```

### BulletinBoardScreen: стили

```javascript

searchInput: { // Этот стиль остается, он используется в HomeScreen

// ...

},

bulletinSearchInput: { // Переименовали для BulletinBoardScreen

height: 45,

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

paddingHorizontal: 15,

fontSize: 16,

borderWidth: 1,

borderColor: '#e2e8f0',

},

modalContainer: { // Этот стиль остается (для галереи в HomeScreen)

// ...

},

addModalContainer: { // Переименовали для модального окна добавления объявления

flex: 1,

justifyContent: 'center',

alignItems: 'center',

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

},

modalTitle: { // Этот стиль остается (для галереи)

// ...

},

addModalTitle: { // Переименовали для модального окна добавления объявления

fontSize: 22,

fontWeight: 'bold',

textAlign: 'center',

marginBottom: 20,

color: '#2d3748',

},

```

### BulletinBoardScreen: использование стилей

```jsx

<View style={styles.searchContainer}>

<TextInput

style={styles.bulletinSearchInput} // Используем переименованный стиль

// ... остальные свойства

/>

</View>

// ...

<Modal

// ... свойства

>

<View style={styles.addModalContainer}> {/\* Переименованный стиль \*/}

<View style={styles.modalContent}>

<Text style={styles.addModalTitle}>Добавить объявление</Text> {/\* Переименованный стиль \*/}

{/\* ... \*/}

</View>

</View>

</Modal>

```

Таким образом, мы устранили все предупреждения ESLint.

Теперь полный исправленный код (сокращенно, только измененные части) выглядит так:

### HomeScreen (useEffect)

```javascript

useEffect(() => {

const timer = setInterval(() => {

if (scrollViewRef.current) {

currentOffset.current += 1;

scrollViewRef.current.scrollTo({

x: currentOffset.current,

animated: false,

});

const maxScroll = ((width \* 0.65) / 4 + 8) \* partnerPhotos.length;

if (currentOffset.current >= maxScroll) {

currentOffset.current = 0;

scrollViewRef.current.scrollTo({ x: 0, animated: false });

}

}

}, 30);

return () => clearInterval(timer);

}, [partnerPhotos.length]); // Добавлена зависимость

```

### Удаление неиспользуемого стиля

В объекте стилей удаляем ключ `detailIcon`.

### ActivitiesScreen (renderFilterModal)

```jsx

const renderFilterModal = () => (

<View style={styles.filterModal}>

<Text style={styles.filterModalTitle}>Фильтры</Text>

<View style={styles.filterSection}>

<Text style={styles.filterSectionLabel}>Сложность</Text> {/\* Изменено \*/}

{/\* ... \*/}

</View>

{/\* ... \*/}

</View>

);

```

### Стили (в конце файла)

```javascript

const styles = StyleSheet.create({

// ... другие стили

// Удален detailIcon

// Для ActivitiesScreen

filterSectionLabel: { // Новое имя для filterLabel в ActivitiesScreen

fontSize: 16,

fontWeight: '600',

marginBottom: 10,

color: '#2d3748',

},

// Для BulletinBoardScreen

bulletinSearchInput: { // Переименованный searchInput

height: 45,

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

paddingHorizontal: 15,

fontSize: 16,

borderWidth: 1,

borderColor: '#e2e8f0',

},

addModalContainer: { // Переименованный modalContainer

flex: 1,

justifyContent: 'center',

alignItems: 'center',

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

},

addModalTitle: { // Переименованный modalTitle

fontSize: 22,

fontWeight: 'bold',

textAlign: 'center',

marginBottom: 20,

color: '#2d3748',

},

// ... остальные стили

});

```

### BulletinBoardScreen (компонент)

```jsx

// В компоненте BulletinBoardScreen:

<View style={styles.searchContainer}>

<TextInput

style={styles.bulletinSearchInput} // Используем переименованный стиль

// ...

/>

</View>

// ...

<Modal

// ...

>

<View style={styles.addModalContainer}> {/\* Переименованный стиль \*/}

<View style={styles.modalContent}>

<Text style={styles.addModalTitle}>Добавить объявление</Text> {/\* Переименованный стиль \*/}

{/\* ... \*/}

</View>

</View>

</Modal>

```

Эти изменения должны устранить все указанные предупреждения ESLint.

Вот исправления для указанных ошибок:

1. Проблема с зависимостью useEffect

javascript

// Было

useEffect(() => {

const timer = setInterval(() => {

// ...

}, 30);

return () => clearInterval(timer);

}, []); // Пустой массив зависимостей

// Стало

useEffect(() => {

const timer = setInterval(() => {

// ...

}, 30);

return () => clearInterval(timer);

}, [partnerPhotos.length]); // Добавили зависимость

2. Удаление неиспользуемого стиля

javascript

// Удалите этот стиль из StyleSheet.create

// detailIcon: {

// fontSize: 18,

// marginRight: 10,

// minWidth: 24,

// },

3-6. Устранение дубликатов ключей стилей

Переименуйте дублирующиеся стили, добавив префиксы:

javascript

const styles = StyleSheet.create({

// Было два filterLabel - переименуем второй

filterLabel: {

// оригинальный стиль

},

filterSectionLabel: { // переименованный

fontSize: 16,

fontWeight: '600',

marginBottom: 10,

color: '#2d3748',

},

// Было два searchInput - переименуем второй

searchInput: {

// оригинальный стиль

},

bulletinSearchInput: { // переименованный

height: 45,

backgroundColor: 'rgba(255, 255, 255, 0.85)',

borderRadius: 12,

paddingHorizontal: 15,

fontSize: 16,

borderWidth: 1,

borderColor: '#e2e8f0',

},

// Было два modalContainer - переименуем второй

modalContainer: {

// оригинальный стиль

},

addModalContainer: { // переименованный

flex: 1,

justifyContent: 'center',

alignItems: 'center',

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

},

// Было два modalTitle - переименуем второй

modalTitle: {

// оригинальный стиль

},

addModalTitle: { // переименованный

fontSize: 22,

fontWeight: 'bold',

textAlign: 'center',

marginBottom: 20,

color: '#2d3748',

},

});

Обновленные компоненты с новыми именами стилей

1. **ActivitiesScreen** (где используется filterLabel):

jsx

// В функции renderFilterModal

<View style={styles.filterSection}>

<Text style={styles.filterSectionLabel}>Сложность</Text> {/\* Было filterLabel \*/}

{/\* ... \*/}

</View>

1. **BulletinBoardScreen** (searchInput):

jsx

<TextInput

style={styles.bulletinSearchInput} // Было searchInput

placeholder="Поиск объявлений..."

{/\* ... \*/}

/>

1. **BulletinBoardScreen** (modalContainer и modalTitle):

jsx

<Modal {/\* ... \*/}>

<View style={styles.addModalContainer}> {/\* Было modalContainer \*/}

<View style={styles.modalContent}>

<Text style={styles.addModalTitle}>Добавить объявление</Text> {/\* Было modalTitle \*/}

{/\* ... \*/}

</View>

</View>

</Modal>

Полный исправленный код компонента BulletinBoardScreen (фрагмент)

javascript

const BulletinBoardScreen = ({ navigation }) => {

// ... остальной код без изменений

return (

// ...

<View style={styles.searchContainer}>

<TextInput

style={styles.bulletinSearchInput} // Исправлено здесь

placeholder="Поиск объявлений..."

{/\* ... \*/}

/>

</View>

// ...

<Modal

animationType="slide"

transparent={true}

visible={addModalVisible}

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

>

<View style={styles.addModalContainer}> // Исправлено здесь

<View style={styles.modalContent}>

<Text style={styles.addModalTitle}>Добавить объявление</Text> // Исправлено здесь

{/\* ... \*/}

</View>

</View>

</Modal>

);

};

Итоговые исправления включают:

1. Добавление зависимости partnerPhotos.length в useEffect
2. Удаление неиспользуемого стиля detailIcon
3. Переименование дублирующихся стилей с добавлением префиксов:
   * filterLabel → filterSectionLabel
   * searchInput → bulletinSearchInput
   * modalContainer → addModalContainer
   * modalTitle → addModalTitle
4. Обновление JSX компонентов для использования новых имен стилей

Эти изменения устранят все указанные ошибки ESLint и сохранят функциональность приложения.