Привет мы с тобой будем делать прототип мобильного приложения агрегатор туров

сейчас я скину тебе код для ознакомления.

1) необходимо создать сущности пользователь, админ, тур оператор , и гид с одной- двумя активностями

Вот наши наработки import React, { useState, useEffect } from 'react';

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

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

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

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

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

const Stack = createStackNavigator();

const Tab = createBottomTabNavigator();

// Модель данных пользователя

const initialUserData = {

id: 1,

name: "Иван Петров",

email: "ivan@example.com",

phone: "+7 (999) 123-45-67",

avatar: "https://randomuser.me/api/portraits/men/1.jpg",

bookings: [

{

id: "B001",

tour: "Восхождение на Авачинский вулкан",

date: "15.07.2023",

status: "Подтвержден",

price: "15 000 ₽",

people: 2,

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

},

{

id: "B002",

tour: "Морская рыбалка",

date: "22.08.2023",

status: "Ожидает оплаты",

price: "22 000 ₽",

people: 4,

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

}

],

documents: [

{ id: "D001", name: "Агентский договор", date: "12.05.2023", type: "contract" },

{ id: "D002", name: "Счет на оплату №123", date: "10.07.2023", type: "invoice" },

{ id: "D003", name: "Акт выполненных работ", date: "17.07.2023", type: "act" }

],

stats: {

totalBookings: 8,

totalSpent: "125 400 ₽",

loyaltyLevel: "Gold",

commission: "12%"

},

notifications: [

{ id: "N001", title: "Новое спецпредложение", message: "Раннее бронирование - дополнительная скидка 5%", date: "2 часа назад", read: false },

{ id: "N002", title: "Изменение в расписании", message: "Ваш тур перенесен на 1 час позже", date: "Вчера", read: true }

]

};

// Главный экран ЛК

const DashboardScreen = ({ navigation }) => {

const [user, setUser] = useState(initialUserData);

const [editModalVisible, setEditModalVisible] = useState(false);

const [tempUserData, setTempUserData] = useState({});

useEffect(() => {

setTempUserData({

name: user.name,

email: user.email,

phone: user.phone

});

}, [user]);

const handleSaveProfile = () => {

setUser({

...user,

name: tempUserData.name,

email: tempUserData.email,

phone: tempUserData.phone

});

setEditModalVisible(false);

};

return (

<ScrollView style={styles.container}>

{/\* Шапка профиля \*/}

<View style={styles.profileHeader}>

<Image source={{ uri: user.avatar }} style={styles.avatar} />

<View style={styles.profileInfo}>

<Text style={styles.userName}>{user.name}</Text>

<Text style={styles.userEmail}>{user.email}</Text>

<Text style={styles.userPhone}>{user.phone}</Text>

<View style={styles.loyaltyBadge}>

<Icon name="star" size={16} color="#FFD700" />

<Text style={styles.loyaltyText}>Уровень: {user.stats.loyaltyLevel}</Text>

</View>

</View>

<TouchableOpacity

style={styles.editButton}

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

>

<Icon name="edit" size={20} color="#3498db" />

</TouchableOpacity>

</View>

{/\* Статистика \*/}

<View style={styles.statsContainer}>

<View style={styles.statCard}>

<Text style={styles.statValue}>{user.stats.totalBookings}</Text>

<Text style={styles.statLabel}>Туров</Text>

</View>

<View style={styles.statCard}>

<Text style={styles.statValue}>{user.stats.totalSpent}</Text>

<Text style={styles.statLabel}>Потрачено</Text>

</View>

<View style={styles.statCard}>

<Text style={styles.statValue}>{user.stats.commission}</Text>

<Text style={styles.statLabel}>Комиссия</Text>

</View>

</View>

{/\* Быстрые действия \*/}

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

<View style={styles.actionsContainer}>

<TouchableOpacity

style={styles.actionCard}

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

>

<Icon name="event" size={30} color="#3498db" />

<Text style={styles.actionText}>Бронирования</Text>

</TouchableOpacity>

<TouchableOpacity

style={styles.actionCard}

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

>

<Icon name="attach-money" size={30} color="#27ae60" />

<Text style={styles.actionText}>Финансы</Text>

</TouchableOpacity>

<TouchableOpacity

style={styles.actionCard}

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

>

<Icon name="description" size={30} color="#e74c3c" />

<Text style={styles.actionText}>Документы</Text>

</TouchableOpacity>

<TouchableOpacity

style={styles.actionCard}

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

>

<Icon name="settings" size={30} color="#7f8c8d" />

<Text style={styles.actionText}>Настройки</Text>

</TouchableOpacity>

</View>

{/\* Последние бронирования \*/}

<View style={styles.sectionHeader}>

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

<TouchableOpacity onPress={() => navigation.navigate('Bookings')}>

<Text style={styles.seeAll}>Все</Text>

</TouchableOpacity>

</View>

{user.bookings.slice(0, 2).map(booking => (

<TouchableOpacity

key={booking.id}

style={styles.bookingCard}

onPress={() => navigation.navigate('BookingDetails', { booking })}

>

<Image source={{ uri: booking.image }} style={styles.bookingImage} />

<View style={styles.bookingInfo}>

<Text style={styles.bookingTitle}>{booking.tour}</Text>

<Text style={styles.bookingDate}>{booking.date} • {booking.people} чел.</Text>

<View style={[styles.statusBadge, booking.status === 'Подтвержден' ? styles.statusConfirmed : styles.statusPending]}>

<Text style={styles.statusText}>{booking.status}</Text>

</View>

</View>

<Text style={styles.bookingPrice}>{booking.price}</Text>

</TouchableOpacity>

))}

{/\* Уведомления \*/}

<View style={styles.sectionHeader}>

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

<TouchableOpacity onPress={() => navigation.navigate('Notifications')}>

<Text style={styles.seeAll}>Все</Text>

</TouchableOpacity>

</View>

{user.notifications.filter(n => !n.read).slice(0, 2).map(notification => (

<TouchableOpacity

key={notification.id}

style={styles.notificationCard}

onPress={() => navigation.navigate('NotificationDetails', { notification })}

>

<View style={styles.notificationIcon}>

<Icon name="notifications" size={24} color="#3498db" />

{!notification.read && <View style={styles.unreadBadge} />}

</View>

<View style={styles.notificationContent}>

<Text style={styles.notificationTitle}>{notification.title}</Text>

<Text style={styles.notificationMessage} numberOfLines={1}>{notification.message}</Text>

<Text style={styles.notificationDate}>{notification.date}</Text>

</View>

</TouchableOpacity>

))}

{/\* Модальное окно редактирования профиля \*/}

<Modal

animationType="slide"

transparent={true}

visible={editModalVisible}

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

>

<View style={styles.modalContainer}>

<View style={styles.modalContent}>

<Text style={styles.modalTitle}>Редактировать профиль</Text>

<TextInput

style={styles.input}

placeholder="Имя"

value={tempUserData.name}

onChangeText={text => setTempUserData({...tempUserData, name: text})}

/>

<TextInput

style={styles.input}

placeholder="Email"

keyboardType="email-address"

value={tempUserData.email}

onChangeText={text => setTempUserData({...tempUserData, email: text})}

/>

<TextInput

style={styles.input}

placeholder="Телефон"

keyboardType="phone-pad"

value={tempUserData.phone}

onChangeText={text => setTempUserData({...tempUserData, phone: text})}

/>

<View style={styles.modalButtons}>

<TouchableOpacity

style={styles.cancelButton}

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

>

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

</TouchableOpacity>

<TouchableOpacity

style={styles.saveButton}

onPress={handleSaveProfile}

>

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

</TouchableOpacity>

</View>

</View>

</View>

</Modal>

</ScrollView>

);

};

// Экран бронирований

const BookingsScreen = ({ navigation }) => {

const user = initialUserData;

const renderBookingItem = ({ item }) => (

<TouchableOpacity

style={styles.bookingCard}

onPress={() => navigation.navigate('BookingDetails', { booking: item })}

>

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

<View style={styles.bookingInfo}>

<Text style={styles.bookingTitle}>{item.tour}</Text>

<Text style={styles.bookingDate}>{item.date} • {item.people} чел.</Text>

<View style={[styles.statusBadge, item.status === 'Подтвержден' ? styles.statusConfirmed : styles.statusPending]}>

<Text style={styles.statusText}>{item.status}</Text>

</View>

</View>

<Text style={styles.bookingPrice}>{item.price}</Text>

</TouchableOpacity>

);

return (

<View style={styles.container}>

<View style={styles.screenHeader}>

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

<TouchableOpacity>

<Icon name="filter-list" size={24} color="#3498db" />

</TouchableOpacity>

</View>

<FlatList

data={user.bookings}

renderItem={renderBookingItem}

keyExtractor={item => item.id}

contentContainerStyle={styles.listContent}

ListEmptyComponent={

<View style={styles.emptyContainer}>

<Icon name="event-busy" size={60} color="#bdc3c7" />

<Text style={styles.emptyText}>У вас пока нет бронирований</Text>

<TouchableOpacity

style={styles.primaryButton}

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

>

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

</TouchableOpacity>

</View>

}

/>

</View>

);

};

// Экран финансов

const FinanceScreen = () => {

const transactions = [

{ id: 'T001', date: '15.07.2023', amount: '-15 000 ₽', description: 'Оплата тура', status: 'Завершено' },

{ id: 'T002', date: '10.07.2023', amount: '+1 800 ₽', description: 'Возврат комиссии', status: 'Завершено' },

{ id: 'T003', date: '05.07.2023', amount: '-22 000 ₽', description: 'Предоплата', status: 'Ожидание' },

{ id: 'T004', date: '01.07.2023', amount: '+3 300 ₽', description: 'Агентское вознаграждение', status: 'Завершено' },

];

return (

<ScrollView style={styles.container}>

<View style={styles.screenHeader}>

<Text style={styles.screenTitle}>Финансы</Text>

<TouchableOpacity>

<Icon name="history" size={24} color="#3498db" />

</TouchableOpacity>

</View>

<View style={styles.balanceCard}>

<Text style={styles.balanceTitle}>Текущий баланс</Text>

<Text style={styles.balanceAmount}>+12 450 ₽</Text>

<Text style={styles.balanceHint}>Доступно для вывода</Text>

<View style={styles.balanceButtons}>

<TouchableOpacity style={styles.outlineButton}>

<Text style={styles.outlineButtonText}>Пополнить</Text>

</TouchableOpacity>

<TouchableOpacity style={styles.primaryButton}>

<Text style={styles.primaryButtonText}>Вывести</Text>

</TouchableOpacity>

</View>

</View>

<Text style={styles.sectionTitle}>История операций</Text>

{transactions.map(transaction => (

<View key={transaction.id} style={styles.transactionCard}>

<View style={styles.transactionInfo}>

<Text style={styles.transactionDate}>{transaction.date}</Text>

<Text style={styles.transactionDesc}>{transaction.description}</Text>

<Text style={[

styles.transactionStatus,

transaction.status === 'Завершено' ? styles.statusSuccess : styles.statusWarning

]}>

{transaction.status}

</Text>

</View>

<Text style={[

styles.transactionAmount,

transaction.amount.startsWith('+') ? styles.positiveAmount : styles.negativeAmount

]}>

{transaction.amount}

</Text>

</View>

))}

</ScrollView>

);

};

// Экран документов

const DocumentsScreen = () => {

const user = initialUserData;

const getDocumentIcon = (type) => {

switch(type) {

case 'contract': return 'description';

case 'invoice': return 'receipt';

case 'act': return 'assignment';

default: return 'insert-drive-file';

}

};

const getDocumentColor = (type) => {

switch(type) {

case 'contract': return '#3498db';

case 'invoice': return '#27ae60';

case 'act': return '#e74c3c';

default: return '#7f8c8d';

}

};

return (

<View style={styles.container}>

<View style={styles.screenHeader}>

<Text style={styles.screenTitle}>Документы</Text>

<TouchableOpacity>

<Icon name="sort" size={24} color="#3498db" />

</TouchableOpacity>

</View>

<FlatList

data={user.documents}

renderItem={({ item }) => (

<TouchableOpacity style={styles.documentCard}>

<View style={[styles.documentIcon, { backgroundColor: getDocumentColor(item.type) + '20' }]}>

<Icon

name={getDocumentIcon(item.type)}

size={28}

color={getDocumentColor(item.type)}

/>

</View>

<View style={styles.documentInfo}>

<Text style={styles.documentName}>{item.name}</Text>

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

</View>

<TouchableOpacity style={styles.downloadButton}>

<Icon name="cloud-download" size={24} color="#3498db" />

</TouchableOpacity>

</TouchableOpacity>

)}

keyExtractor={item => item.id}

contentContainerStyle={styles.listContent}

ListEmptyComponent={

<View style={styles.emptyContainer}>

<Icon name="folder-open" size={60} color="#bdc3c7" />

<Text style={styles.emptyText}>Документы отсутствуют</Text>

</View>

}

/>

</View>

);

};

// Экран настроек

const SettingsScreen = ({ navigation }) => {

const [notificationsEnabled, setNotificationsEnabled] = useState(true);

const settingsOptions = [

{ icon: 'notifications', title: 'Уведомления', action: () => setNotificationsEnabled(!notificationsEnabled) },

{ icon: 'lock', title: 'Безопасность', action: () => navigation.navigate('Security') },

{ icon: 'language', title: 'Язык', action: () => navigation.navigate('Language') },

{ icon: 'help', title: 'Помощь', action: () => Linking.openURL('https://zimaletokamchatka.com/support') },

{ icon: 'info', title: 'О приложении', action: () => navigation.navigate('About') },

{ icon: 'exit-to-app', title: 'Выйти', action: () => navigation.navigate('Login') },

];

return (

<ScrollView style={styles.container}>

<Text style={styles.screenTitle}>Настройки</Text>

<View style={styles.settingsGroup}>

{settingsOptions.map((item, index) => (

<TouchableOpacity

key={index}

style={styles.settingItem}

onPress={item.action}

>

<View style={styles.settingLeft}>

<Icon name={item.icon} size={24} color="#3498db" style={styles.settingIcon} />

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

</View>

{item.icon === 'notifications' ? (

<View style={styles.toggleContainer}>

<Text style={styles.toggleText}>{notificationsEnabled ? 'Вкл' : 'Выкл'}</Text>

<TouchableOpacity

style={[styles.toggle, notificationsEnabled && styles.toggleActive]}

onPress={() => setNotificationsEnabled(!notificationsEnabled)}

>

<View style={[styles.toggleCircle, notificationsEnabled && styles.toggleCircleActive]} />

</TouchableOpacity>

</View>

) : (

<Icon name="chevron-right" size={24} color="#bdc3c7" />

)}

</TouchableOpacity>

))}

</View>

<View style={styles.settingsGroup}>

<Text style={styles.groupTitle}>Обратная связь</Text>

<TouchableOpacity style={styles.settingItem}>

<View style={styles.settingLeft}>

<Icon name="star" size={24} color="#FFD700" style={styles.settingIcon} />

<Text style={styles.settingText}>Оценить приложение</Text>

</View>

<Icon name="chevron-right" size={24} color="#bdc3c7" />

</TouchableOpacity>

<TouchableOpacity

style={styles.settingItem}

onPress={() => Linking.openURL('mailto:support@zimaletokamchatka.com')}

>

<View style={styles.settingLeft}>

<Icon name="mail" size={24} color="#e74c3c" style={styles.settingIcon} />

<Text style={styles.settingText}>Написать в поддержку</Text>

</View>

<Icon name="chevron-right" size={24} color="#bdc3c7" />

</TouchableOpacity>

</View>

</ScrollView>

);

};

// Навигация ЛК

const LKNavigator = () => (

<Tab.Navigator

screenOptions={({ route }) => ({

tabBarIcon: ({ focused, color, size }) => {

let iconName;

if (route.name === 'Dashboard') iconName = 'home';

else if (route.name === 'Bookings') iconName = 'event';

else if (route.name === 'Finance') iconName = 'attach-money';

else if (route.name === 'Documents') iconName = 'folder';

else if (route.name === 'Settings') iconName = 'settings';

return <Icon name={iconName} size={size} color={color} />;

},

tabBarActiveTintColor: '#3498db',

tabBarInactiveTintColor: '#95a5a6',

tabBarStyle: {

paddingBottom: 5,

height: 60

},

tabBarLabelStyle: {

fontSize: 12,

marginBottom: 5

},

headerShown: false

})}

>

<Tab.Screen name="Dashboard" component={DashboardScreen} options={{ title: 'Главная' }} />

<Tab.Screen name="Bookings" component={BookingsScreen} options={{ title: 'Бронирования' }} />

<Tab.Screen name="Finance" component={FinanceScreen} options={{ title: 'Финансы' }} />

<Tab.Screen name="Documents" component={DocumentsScreen} options={{ title: 'Документы' }} />

<Tab.Screen name="Settings" component={SettingsScreen} options={{ title: 'Настройки' }} />

</Tab.Navigator>

);

// Главный компонент приложения

export default function App() {

return (

<NavigationContainer>

<Stack.Navigator>

<Stack.Screen

name="LKNavigator"

component={LKNavigator}

options={{ headerShown: false }}

/>

{/\* Здесь можно добавить другие экраны \*/}

</Stack.Navigator>

</NavigationContainer>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

padding: 16,

backgroundColor: '#f8f9fa',

},

profileHeader: {

flexDirection: 'row',

alignItems: 'center',

marginBottom: 20,

},

avatar: {

width: 80,

height: 80,

borderRadius: 40,

marginRight: 16,

},

profileInfo: {

flex: 1,

},

userName: {

fontSize: 20,

fontWeight: 'bold',

color: '#2c3e50',

},

userEmail: {

fontSize: 14,

color: '#7f8c8d',

marginTop: 4,

},

userPhone: {

fontSize: 14,

color: '#7f8c8d',

marginTop: 2,

},

loyaltyBadge: {

flexDirection: 'row',

alignItems: 'center',

backgroundColor: '#fef9e7',

paddingVertical: 4,

paddingHorizontal: 8,

borderRadius: 12,

alignSelf: 'flex-start',

marginTop: 8,

},

loyaltyText: {

fontSize: 12,

color: '#f39c12',

marginLeft: 4,

},

editButton: {

padding: 8,

},

statsContainer: {

flexDirection: 'row',

justifyContent: 'space-between',

marginBottom: 20,

},

statCard: {

backgroundColor: 'white',

borderRadius: 12,

padding: 16,

alignItems: 'center',

width: '30%',

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

statValue: {

fontSize: 18,

fontWeight: 'bold',

color: '#3498db',

},

statLabel: {

fontSize: 12,

color: '#7f8c8d',

marginTop: 4,

},

sectionTitle: {

fontSize: 18,

fontWeight: 'bold',

color: '#2c3e50',

marginBottom: 12,

},

actionsContainer: {

flexDirection: 'row',

flexWrap: 'wrap',

justifyContent: 'space-between',

marginBottom: 20,

},

actionCard: {

backgroundColor: 'white',

borderRadius: 12,

padding: 16,

alignItems: 'center',

width: '48%',

marginBottom: 12,

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

actionText: {

fontSize: 14,

marginTop: 8,

color: '#2c3e50',

},

sectionHeader: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

marginBottom: 12,

},

seeAll: {

color: '#3498db',

fontSize: 14,

},

bookingCard: {

backgroundColor: 'white',

borderRadius: 12,

padding: 16,

marginBottom: 12,

flexDirection: 'row',

alignItems: 'center',

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

bookingImage: {

width: 60,

height: 60,

borderRadius: 8,

marginRight: 12,

},

bookingInfo: {

flex: 1,

},

bookingTitle: {

fontSize: 16,

fontWeight: '600',

color: '#2c3e50',

},

bookingDate: {

fontSize: 14,

color: '#7f8c8d',

marginTop: 4,

},

statusBadge: {

alignSelf: 'flex-start',

paddingVertical: 4,

paddingHorizontal: 8,

borderRadius: 12,

marginTop: 6,

},

statusConfirmed: {

backgroundColor: '#e8f8f5',

},

statusPending: {

backgroundColor: '#fef9e7',

},

statusText: {

fontSize: 12,

fontWeight: '500',

},

bookingPrice: {

fontSize: 16,

fontWeight: 'bold',

color: '#2c3e50',

},

notificationCard: {

backgroundColor: 'white',

borderRadius: 12,

padding: 16,

marginBottom: 12,

flexDirection: 'row',

alignItems: 'center',

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

notificationIcon: {

marginRight: 16,

position: 'relative',

},

unreadBadge: {

position: 'absolute',

top: -2,

right: -2,

width: 10,

height: 10,

borderRadius: 5,

backgroundColor: '#e74c3c',

},

notificationContent: {

flex: 1,

},

notificationTitle: {

fontSize: 16,

fontWeight: '600',

color: '#2c3e50',

},

notificationMessage: {

fontSize: 14,

color: '#7f8c8d',

marginTop: 4,

},

notificationDate: {

fontSize: 12,

color: '#bdc3c7',

marginTop: 4,

},

modalContainer: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

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

},

modalContent: {

width: '90%',

backgroundColor: 'white',

borderRadius: 16,

padding: 24,

shadowColor: '#000',

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

shadowOpacity: 0.3,

shadowRadius: 8,

elevation: 5,

},

modalTitle: {

fontSize: 20,

fontWeight: 'bold',

marginBottom: 20,

textAlign: 'center',

color: '#2c3e50',

},

input: {

height: 50,

borderColor: '#e0e0e0',

borderWidth: 1,

borderRadius: 10,

marginBottom: 16,

paddingHorizontal: 16,

fontSize: 16,

},

modalButtons: {

flexDirection: 'row',

justifyContent: 'space-between',

marginTop: 10,

},

cancelButton: {

flex: 1,

backgroundColor: '#e0e0e0',

borderRadius: 10,

padding: 14,

alignItems: 'center',

marginRight: 10,

},

cancelButtonText: {

fontSize: 16,

fontWeight: '600',

color: '#7f8c8d',

},

saveButton: {

flex: 1,

backgroundColor: '#3498db',

borderRadius: 10,

padding: 14,

alignItems: 'center',

},

saveButtonText: {

fontSize: 16,

fontWeight: '600',

color: 'white',

},

screenHeader: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

marginBottom: 20,

},

screenTitle: {

fontSize: 24,

fontWeight: 'bold',

color: '#2c3e50',

},

listContent: {

paddingBottom: 20,

},

emptyContainer: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

padding: 40,

},

emptyText: {

fontSize: 16,

color: '#bdc3c7',

marginTop: 16,

textAlign: 'center',

},

primaryButton: {

backgroundColor: '#3498db',

borderRadius: 10,

padding: 14,

alignItems: 'center',

marginTop: 20,

},

primaryButtonText: {

fontSize: 16,

fontWeight: '600',

color: 'white',

},

balanceCard: {

backgroundColor: '#3498db',

borderRadius: 16,

padding: 24,

marginBottom: 24,

},

balanceTitle: {

fontSize: 16,

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

marginBottom: 8,

},

balanceAmount: {

fontSize: 32,

fontWeight: 'bold',

color: 'white',

},

balanceHint: {

fontSize: 14,

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

marginTop: 4,

},

balanceButtons: {

flexDirection: 'row',

justifyContent: 'space-between',

marginTop: 24,

},

outlineButton: {

flex: 1,

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

borderRadius: 10,

padding: 14,

alignItems: 'center',

marginRight: 10,

},

outlineButtonText: {

fontSize: 16,

fontWeight: '600',

color: 'white',

},

transactionCard: {

backgroundColor: 'white',

borderRadius: 12,

padding: 16,

marginBottom: 12,

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

transactionInfo: {

flex: 1,

},

transactionDate: {

fontSize: 12,

color: '#7f8c8d',

},

transactionDesc: {

fontSize: 16,

color: '#2c3e50',

marginTop: 4,

},

transactionStatus: {

fontSize: 12,

marginTop: 4,

paddingVertical: 2,

paddingHorizontal: 8,

borderRadius: 10,

alignSelf: 'flex-start',

},

statusSuccess: {

backgroundColor: '#e8f8f5',

color: '#27ae60',

},

statusWarning: {

backgroundColor: '#fef9e7',

color: '#f39c12',

},

transactionAmount: {

fontSize: 16,

fontWeight: 'bold',

},

positiveAmount: {

color: '#27ae60',

},

negativeAmount: {

color: '#e74c3c',

},

documentCard: {

backgroundColor: 'white',

borderRadius: 12,

padding: 16,

marginBottom: 12,

flexDirection: 'row',

alignItems: 'center',

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

documentIcon: {

width: 50,

height: 50,

borderRadius: 10,

justifyContent: 'center',

alignItems: 'center',

marginRight: 16,

},

documentInfo: {

flex: 1,

},

documentName: {

fontSize: 16,

fontWeight: '600',

color: '#2c3e50',

},

documentDate: {

fontSize: 14,

color: '#7f8c8d',

marginTop: 4,

},

downloadButton: {

padding: 8,

},

settingsGroup: {

backgroundColor: 'white',

borderRadius: 12,

marginBottom: 24,

shadowColor: '#000',

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

shadowOpacity: 0.1,

shadowRadius: 4,

elevation: 3,

},

groupTitle: {

fontSize: 14,

fontWeight: '600',

color: '#7f8c8d',

padding: 16,

paddingBottom: 8,

},

settingItem: {

flexDirection: 'row',

justifyContent: 'space-between',

alignItems: 'center',

padding: 16,

borderBottomWidth: 1,

borderBottomColor: '#f0f0f0',

},

settingLeft: {

flexDirection: 'row',

alignItems: 'center',

},

settingIcon: {

marginRight: 16,

},

settingText: {

fontSize: 16,

color: '#2c3e50',

},

toggleContainer: {

flexDirection: 'row',

alignItems: 'center',

},

toggleText: {

fontSize: 14,

color: '#7f8c8d',

marginRight: 8,

},

toggle: {

width: 50,

height: 28,

borderRadius: 14,

backgroundColor: '#e0e0e0',

justifyContent: 'center',

paddingHorizontal: 3,

},

toggleActive: {

backgroundColor: '#3498db',

},

toggleCircle: {

width: 22,

height: 22,

borderRadius: 11,

backgroundColor: 'white',

},

toggleCircleActive: {

transform: [{ translateX: 22 }],

},

});