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

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

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

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

import AsyncStorage from '@react-native-async-storage/async-storage';

const Tab = createBottomTabNavigator();

const JobsScreen = ({ navigation }) => {

const [jobs, setJobs] = useState([]);

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

const [page, setPage] = useState(1);

const [error, setError] = useState(null);

const [bookmarkedJobs, setBookmarkedJobs] = useState([]);

const fetchJobs = async () => {

try {

const response = await fetch(`https://testapi.getlokalapp.com/common/jobs?page=${page}`);

const data = await response.json();

setJobs(prevJobs => [...prevJobs, ...data.jobs]);

setLoading(false);

} catch (err) {

setError(err.message);

setLoading(false);

}

};

useEffect(() => {

fetchJobs();

}, [page]);

const handleLoadMore = () => {

if (!loading) {

setPage(prevPage => prevPage + 1);

}

};

const handleBookmark = async (job) => {

const updatedBookmarks = [...bookmarkedJobs, job];

setBookmarkedJobs(updatedBookmarks);

await AsyncStorage.setItem('bookmarkedJobs', JSON.stringify(updatedBookmarks));

};

const renderJob = ({ item }) => (

<TouchableOpacity onPress={() => navigation.navigate('JobDetails', { job: item })}>

<View style={styles.card}>

<Text>{item.title}</Text>

<Text>{item.location}</Text>

<Text>{item.salary}</Text>

<Text>{item.phone}</Text>

<TouchableOpacity onPress={() => handleBookmark(item)}>

<Text>Bookmark</Text>

</TouchableOpacity>

</View>

</TouchableOpacity>

);

return (

<View>

{loading ? <Text>Loading...</Text> : error ? <Text>{error}</Text> : (

<FlatList

data={jobs}

renderItem={renderJob}

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

onEndReached={handleLoadMore}

onEndReachedThreshold={0.5}

/>

)}

</View>

);

};

const BookmarksScreen = () => {

const [bookmarkedJobs, setBookmarkedJobs] = useState([]);

useEffect(() => {

const loadBookmarks = async () => {

const storedBookmarks = await AsyncStorage.getItem('bookmarkedJobs');

if (storedBookmarks) {

setBookmarkedJobs(JSON.parse(storedBookmarks));

}

};

loadBookmarks();

}, []);

return (

<View>

{bookmarkedJobs.length === 0 ? <Text>No Bookmarks</Text> : (

bookmarkedJobs.map(job => (

<View key={job.id} style={styles.card}>

<Text>{job.title}</Text>

<Text>{job.location}</Text>

<Text>{job.salary}</Text>

<Text>{job.phone}</Text>

</View>

))

)}

</View>

);

};

const JobDetailsScreen = ({ route }) => {

const { job } = route.params;

return (

<View style={styles.card}>

<Text>{job.title}</Text>

<Text>{job.location}</Text>

<Text>{job.salary}</Text>

<Text>{job.phone}</Text>

<Text>{job.description}</Text>

</View>

);

};

export default function App() {

return (

<NavigationContainer>

<Tab.Navigator>

<Tab.Screen name="Jobs" component={JobsScreen} />

<Tab.Screen name="Bookmarks" component={BookmarksScreen} />

</Tab.Navigator>

</NavigationContainer>

);

}

const styles = StyleSheet.create({

card: {

padding: 10,

margin: 10,

borderWidth: 1,

borderColor: '#ccc',

borderRadius: 5,

},

});