// App.js

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

import { View, Text, TextInput, Button, StyleSheet, Alert, ActivityIndicator } from 'react-native';

import { Picker } from '@react-native-picker/picker';

import axios from 'axios';

export default function App() {

const [amount, setAmount] = useState('');

const [fromCurrency, setFromCurrency] = useState('USD');

const [toCurrency, setToCurrency] = useState('EUR');

const [currencies, setCurrencies] = useState([]);

const [convertedAmount, setConvertedAmount] = useState(null);

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

const API\_URL = 'https://api.exchangerate-api.com/v4/latest/';

useEffect(() => {

fetchCurrencies();

}, []);

const fetchCurrencies = async () => {

try {

setLoading(true);

const response = await axios.get(`${API\_URL}USD`);

setCurrencies(Object.keys(response.data.rates));

} catch (error) {

Alert.alert('Error', 'Failed to fetch currency data');

} finally {

setLoading(false);

}

};

const handleConvert = async () => {

if (!amount || isNaN(amount)) {

Alert.alert('Invalid input', 'Please enter a valid number');

return;

}

try {

setLoading(true);

const response = await axios.get(`${API\_URL}${fromCurrency}`);

const rate = response.data.rates[toCurrency];

const result = (parseFloat(amount) \* rate).toFixed(2);

setConvertedAmount(`${result} ${toCurrency}`);

} catch (error) {

Alert.alert('Conversion failed', 'Unable to convert currency');

} finally {

setLoading(false);

}

};

return (

<View style={styles.container}>

<Text style={styles.title}>Currency Converter</Text>

<TextInput

style={styles.input}

placeholder="Enter amount"

keyboardType="numeric"

value={amount}

onChangeText={setAmount}

/>

<Text>From:</Text>

<Picker

selectedValue={fromCurrency}

style={styles.picker}

onValueChange={(itemValue) => setFromCurrency(itemValue)}>

{currencies.map((currency) => (

<Picker.Item label={currency} value={currency} key={currency} />

))}

</Picker>

<Text>To:</Text>

<Picker

selectedValue={toCurrency}

style={styles.picker}

onValueChange={(itemValue) => setToCurrency(itemValue)}>

{currencies.map((currency) => (

<Picker.Item label={currency} value={currency} key={currency} />

))}

</Picker>

<Button title="Convert" onPress={handleConvert} />

{loading && <ActivityIndicator size="large" color="#0000ff" />}

{convertedAmount && (

<Text style={styles.result}>Converted Amount: {convertedAmount}</Text>

)}

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

padding: 20,

justifyContent: 'center',

backgroundColor: '#f5f5f5',

},

title: {

fontSize: 24,

fontWeight: 'bold',

textAlign: 'center',

marginBottom: 20,

},

input: {

borderWidth: 1,

borderColor: '#ccc',

padding: 10,

marginBottom: 10,

borderRadius: 5,

backgroundColor: '#fff',

},

picker: {

height: 50,

width: '100%',

marginBottom: 20,

},

result: {

marginTop: 20,

fontSize: 18,

textAlign: 'center',

color: 'green',

},

});