// lib/price\_editor.dart  
import 'package:flutter/material.dart';  
import 'package:shared\_preferences/shared\_preferences.dart';  
import 'dart:convert';  
import 'main.dart';  
  
class PriceEditorScreen extends StatefulWidget {  
 final PriceStore? store;  
 const PriceEditorScreen({super.key, this.store});  
  
 @override  
 State<PriceEditorScreen> createState() => \_PriceEditorScreenState();  
}  
  
class \_PriceEditorScreenState extends State<PriceEditorScreen> {  
 Map<String, dynamic> data = {};  
  
 @override  
 void initState() {  
 super.initState();  
 \_load();  
 }  
  
 Future<void> \_load() async {  
 final prefs = await SharedPreferences.getInstance();  
 final s = prefs.getString(PriceStore.\_key);  
 if (s != null) {  
 try {  
 data = json.decode(s);  
 } catch (\_) { data = {}; }  
 }  
 if (data.isEmpty) {  
 final ps = PriceStore();  
 data = ps.data;  
 }  
 setState(() {});  
 }  
  
 Future<void> \_save() async {  
 final prefs = await SharedPreferences.getInstance();  
 await prefs.setString(PriceStore.\_key, json.encode(data));  
 ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('قیمت‌ها ذخیره شدند')));  
 }  
  
 void \_addBlade() {  
 data['blades']['تیغه جدید'] = {"price": 300000, "thickness": 2.0};  
 setState(() {});  
 }  
 void \_addMotor() { data['motors']['موتور جدید'] = 3000000; setState(() {}); }  
 void \_addShaft() { data['shafts']['13'] = 160000; setState(() {}); }  
 void \_addBox() { data['boxes']['6x6'] = 120000; setState(() {}); }  
  
 @override  
 Widget build(BuildContext context) {  
 if (data.isEmpty) return Scaffold(appBar: AppBar(title: const Text('قیمت پایه')), body: const Center(child: CircularProgressIndicator()));  
 return Scaffold(  
 appBar: AppBar(title: const Text('قیمت پایه')),  
 body: Padding(  
 padding: const EdgeInsets.all(12),  
 child: ListView(  
 children: [  
 const Text('تیغه‌ها', style: TextStyle(fontSize: 18)),  
 ...((data['blades'] as Map).keys).map<Widget>((k) {  
 final item = data['blades'][k];  
 return ListTile(  
 title: Text(k),  
 subtitle: Text('قیمت: ${item['price']} تومان - ضخامت: ${item['thickness']} cm'),  
 trailing: IconButton(icon: const Icon(Icons.delete), onPressed: () { data['blades'].remove(k); setState((){}); }),  
 onTap: () async {  
 final res = await showDialog(context: context, builder: (\_) => \_editBladeDialog(k, item));  
 if (res == true) setState((){});  
 },  
 );  
 }).toList(),  
 ElevatedButton(onPressed: \_addBlade, child: const Text('افزودن تیغه')),  
 const SizedBox(height: 12),  
 const Text('موتورها', style: TextStyle(fontSize: 18)),  
 ...((data['motors'] as Map).keys).map<Widget>((k) {  
 return ListTile(  
 title: Text(k),  
 subtitle: Text('قیمت: ${data['motors'][k]} تومان'),  
 trailing: IconButton(icon: const Icon(Icons.delete), onPressed: () { data['motors'].remove(k); setState((){}); }),  
 onTap: () async {  
 final res = await \_editSimpleNumberDialog('موتور', k, data['motors'][k]);  
 if (res == true) setState((){});  
 },  
 );  
 }).toList(),  
 ElevatedButton(onPressed: \_addMotor, child: const Text('افزودن موتور')),  
 const SizedBox(height: 12),  
 const Text('شفت‌ها', style: TextStyle(fontSize: 18)),  
 ...((data['shafts'] as Map).keys).map<Widget>((k) {  
 return ListTile(  
 title: Text(k),  
 subtitle: Text('قیمت: ${data['shafts'][k]} تومان'),  
 trailing: IconButton(icon: const Icon(Icons.delete), onPressed: () { data['shafts'].remove(k); setState((){}); }),  
 onTap: () async { final res = await \_editSimpleNumberDialog('شفت', k, data['shafts'][k]); if (res == true) setState((){}); },  
 );  
 }).toList(),  
 ElevatedButton(onPressed: \_addShaft, child: const Text('افزودن شفت')),  
 const SizedBox(height: 12),  
 const Text('قوطی‌ها', style: TextStyle(fontSize: 18)),  
 ...((data['boxes'] as Map).keys).map<Widget>((k) {  
 return ListTile(  
 title: Text(k),  
 subtitle: Text('قیمت: ${data['boxes'][k]} تومان'),  
 trailing: IconButton(icon: const Icon(Icons.delete), onPressed: () { data['boxes'].remove(k); setState((){}); }),  
 onTap: () async { final res = await \_editSimpleNumberDialog('قوطی', k, data['boxes'][k]); if (res == true) setState((){}); },  
 );  
 }).toList(),  
 ElevatedButton(onPressed: \_addBox, child: const Text('افزودن قوطی')),  
 const SizedBox(height: 12),  
 const Text('هزینه‌های پایه', style: TextStyle(fontSize: 18)),  
 \_buildNumberField('نصب (تومان در متر مربع)', 'install'),  
 \_buildNumberField('جوشکاری (تومان)', 'welding'),  
 \_buildNumberField('حمل (تومان)', 'transport'),  
 \_buildNumberField('جاقفلی ساده (تومان)', 'lock\_simple'),  
 \_buildNumberField('جاقفلی برقی (تومان)', 'lock\_electric'),  
 \_buildNumberField('کاور موتور (تومان)', 'motor\_cover'),  
 const SizedBox(height: 20),  
 ElevatedButton.icon(onPressed: \_save, icon: const Icon(Icons.save), label: const Text('ذخیره')),  
 ],  
 ),  
 ),  
 );  
 }  
  
 Widget \_buildNumberField(String label, String key) {  
 return ListTile(  
 title: Text(label),  
 trailing: SizedBox(  
 width: 140,  
 child: TextFormField(  
 initialValue: data[key].toString(),  
 keyboardType: const TextInputType.numberWithOptions(decimal: false),  
 onChanged: (v) { data[key] = int.tryParse(v) ?? data[key]; },  
 decoration: const InputDecoration(border: OutlineInputBorder()),  
 ),  
 ),  
 );  
 }  
  
 Widget \_editBladeDialog(String key, dynamic item) {  
 final nameCtrl = TextEditingController(text: key);  
 final priceCtrl = TextEditingController(text: item['price'].toString());  
 final thicknessCtrl = TextEditingController(text: item['thickness'].toString());  
 return AlertDialog(  
 title: const Text('ویرایش تیغه'),  
 content: Column(  
 mainAxisSize: MainAxisSize.min,  
 children: [  
 TextField(controller: nameCtrl, decoration: const InputDecoration(labelText: 'نام')),  
 TextField(controller: priceCtrl, decoration: const InputDecoration(labelText: 'قیمت'), keyboardType: TextInputType.number),  
 TextField(controller: thicknessCtrl, decoration: const InputDecoration(labelText: 'ضخامت (cm)'), keyboardType: TextInputType.number),  
 ],  
 ),  
 actions: [  
 TextButton(onPressed: () { Navigator.pop(context, false); }, child: const Text('انصراف')),  
 TextButton(onPressed: () {  
 final newName = nameCtrl.text;  
 final newPrice = int.tryParse(priceCtrl.text) ?? item['price'];  
 final newThickness = double.tryParse(thicknessCtrl.text) ?? item['thickness'];  
 if (newName != key) {  
 final value = data['blades'][key];  
 data['blades'].remove(key);  
 data['blades'][newName] = {"price": newPrice, "thickness": newThickness};  
 } else {  
 data['blades'][key] = {"price": newPrice, "thickness": newThickness};  
 }  
 Navigator.pop(context, true);  
 }, child: const Text('ذخیره')),  
 ],  
 );  
 }  
  
 Future<bool?> \_editSimpleNumberDialog(String title, String key, dynamic value) {  
 final ctrl = TextEditingController(text: value.toString());  
 return showDialog<bool>(context: context, builder: (\_) {  
 return AlertDialog(  
 title: Text('ویرایش $title'),  
 content: TextField(controller: ctrl, keyboardType: TextInputType.number, decoration: const InputDecoration(labelText: 'قیمت')),  
 actions: [  
 TextButton(onPressed: () => Navigator.pop(context, false), child: const Text('انصراف')),  
 TextButton(onPressed: () {  
 final v = int.tryParse(ctrl.text) ?? value;  
 if (title == 'موتور') data['motors'][key] = v;  
 else if (title == 'شفت') data['shafts'][key] = v;  
 else if (title == 'قوطی') data['boxes'][key] = v;  
 Navigator.pop(context, true);  
 }, child: const Text('ذخیره')),  
 ],  
 );  
 });  
 }  
}