

CODE MAIN.DART

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:convert';
import 'dart:async';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
    const MyApp({super.key});
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            debugShowCheckedModeBanner: false,
            theme: ThemeData(
                primarySwatch: Colors.blue,
                useMaterial3: true,
                scaffoldBackgroundColor: const Color(0xFFFF0F4F8),
            ),
            initialRoute: '/',
            routes: {
                '/': (context) => const LoginPage(),
                '/register': (context) => const RegisterPage(),
                '/dashboard': (context) => const Dashboard(),
            },
        );
    }
}

// --- 1. MÀN HÌNH ĐĂNG NHẬP ---
class LoginPage extends StatefulWidget {
    const LoginPage({super.key});
    @override
    State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
    final TextEditingController _mssvController = TextEditingController();
    final TextEditingController _passController = TextEditingController();
    bool _isLoading = false;

    void _handleLogin() async {
        String mssv = _mssvController.text.trim();
```

```
String pass = _passController.text.trim();
if (mssv.isEmpty || pass.isEmpty) return;

setState(() => _isLoading = true);
final prefs = await SharedPreferences.getInstance();
String? savedMssv = prefs.getString('saved_mssv');
String? savedPass = prefs.getString('saved_pass');
await Future.delayed(const Duration(milliseconds: 600));
setState(() => _isLoading = false);

if ((mssv == savedMssv && pass == savedPass) || (mssv == "123" && pass ==
"123")) {
    if (!mounted) return;
    Navigator.pushReplacementNamed(context, '/dashboard');
} else {
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content:
Text("Sai MSSV hoặc mật khẩu!")));
}
}

@Override
Widget build(BuildContext context) {
return Scaffold(
body: Container(
width: double.infinity,
decoration: const BoxDecoration(
gradient: LinearGradient(
begin: Alignment.topCenter,
colors: [Color(0xFF2196F3), Color(0xFFE3F2FD)],
),
),
),
child: Column(
children: [
const SizedBox(height: 60),
const Icon(Icons.school_rounded, size: 80, color: Colors.white),
const Text("TRƯỜNG CƠ CÔNG THƯƠNG TP.HCM", style: TextStyle(color:
Colors.white, fontWeight: FontWeight.bold, fontSize: 16)),
const Text("KHOA: ĐIỆN - ĐIỆN TỬ", style: TextStyle(color:
Colors.white70, fontSize: 13)),
const SizedBox(height: 30),
Expanded(
child: Container(
padding: const EdgeInsets.symmetric(horizontal: 30, vertical:
30),
decoration: const BoxDecoration(

```

```
        color: Colors.white,
        borderRadius: BorderRadius.only(topLeft: Radius.circular(50),
topRight: Radius.circular(50)),
    ),
    child: SingleChildScrollView(
        child: Column(
            children: [
                const Text("MÔ HÌNH ĐẾM PHÂN LOẠI SẢN PHẨM", style:
TextStyle(fontSize: 24, fontWeight: FontWeight.w900, color: Colors.blue)),
                const SizedBox(height: 10),
                const Text("GVHD: Nguyễn Kim Suyên", style:
TextStyle(fontSize: 14, fontWeight: FontWeight.bold, color: Colors.blueGrey)),
                const Text("SVTH: Nguyễn Đặng Khoa - Nguyễn Hữu Bách",
textAlign: TextAlign.center, style: TextStyle(fontSize: 13, color: Colors.grey)),
                const SizedBox(height: 35),
                _buildModernInput(_mssvController, "Nhập MSSV",
Icons.person_outline, false),
                const SizedBox(height: 15),
                _buildModernInput(_passController, "Nhập Password",
Icons.lock_outline, true),
                const SizedBox(height: 35),
                SizedBox(
                    width: double.infinity, height: 55,
                    child: ElevatedButton(
                        onPressed: _isLoading ? null : _handleLogin,
                        style: ElevatedButton.styleFrom(
                            backgroundColor: Colors.blueAccent,
                            shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(30)),
                            elevation: 4,
                        ),
                        child: _isLoading
                            ? const CircularProgressIndicator(color:
Colors.white)
                            : const Text("ĐĂNG NHẬP", style: TextStyle(color:
Colors.white, fontSize: 18, fontWeight: FontWeight.bold)),
                    ),
                ),
                TextButton(onPressed: () => Navigator.pushNamed(context,
'/register'), child: const Text("Chưa có tài khoản? Đăng ký ngay")),
            ],
        ),
    ),
),
),
),
),
),
```

```

        ],
        ),
        ),
    );
}

Widget _buildModernInput(TextEditingController controller, String hint,
IconData icon, bool isPass) {
    return Container(
        decoration: BoxDecoration(
            color: Colors.grey[100],
            borderRadius: BorderRadius.circular(30),
            boxShadow: [BoxShadow(color: Colors.black.withOpacity(0.05), blurRadius: 10, offset: const Offset(0, 4))],
        ),
        child: TextField(
            controller: controller, obscureText: isPass,
            decoration: InputDecoration(
                prefixIcon: Icon(icon, color: Colors.blue),
                hintText: hint,
                border: InputBorder.none,
                contentPadding: const EdgeInsets.symmetric(vertical: 18, horizontal: 20),
            ),
        ),
    );
}

// --- 2. MÀN HÌNH ĐĂNG KÝ ---
class RegisterPage extends StatefulWidget {
    const RegisterPage({super.key});
    @override
    State<RegisterPage> createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
    final TextEditingController _mssvReg = TextEditingController();
    final TextEditingController _passReg = TextEditingController();
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: const Text("Đăng ký tài khoản")),
            body: Padding(
                padding: const EdgeInsets.all(30.0),

```

```

        child: Column(
            children: [
                TextField(controller: _mssvReg, decoration: const
InputDecoration(labelText: "Nhập MSSV mới", border: OutlineInputBorder())),
                const SizedBox(height: 15),
                TextField(controller: _passReg, obscureText: true, decoration: const
InputDecoration(labelText: "Nhập Mật khẩu mới", border: OutlineInputBorder())),
                const SizedBox(height: 30),
                SizedBox(
                    width: double.infinity, height: 50,
                    child: ElevatedButton(onPressed: () async {
                        final prefs = await SharedPreferences.getInstance();
                        await prefs.setString('saved_mssv', _mssvReg.text);
                        await prefs.setString('saved_pass', _passReg.text);
                        if (!mounted) return;
                        Navigator.pop(context);
                    }, style: ElevatedButton.styleFrom(backgroundColor:
Colors.blueAccent), child: const Text("XÁC NHẬN", style: TextStyle(color:
Colors.white))),),
                ),
            ],
        ),
    );
}

}

// --- 3. MÀN HÌNH ĐIỀU KHIỂN (DASHBOARD) ---
class Dashboard extends StatefulWidget {
    const Dashboard({super.key});
    @override
    State<Dashboard> createState() => _DashboardState();
}

class _DashboardState extends State<Dashboard> {
    final String espIp = "10.88.1.51";
    int total = 0, red = 0, blue = 0;
    double currentHeight = 0.0;
    bool isLed10n = false, isLed20n = false, isConnected = false;
    Timer? _timer;

    @override
    void initState() {
        super.initState();
        _timer = Timer.periodic(const Duration(seconds: 1), (timer) => fetchData());
    }
}

```

```

}

@Override
void dispose() { _timer?.cancel(); super.dispose(); }

Future<void> fetchData() async {
    try {
        final response = await
http.get(Uri.parse('http://$espIp/data')).timeout(const Duration(milliseconds:
900));
        if (response.statusCode == 200) {
            final Map<String, dynamic> data = json.decode(response.body);
            setState(() {
                total = data['total'] ?? 0;
                red = data['red'] ?? 0;
                blue = data['blue'] ?? 0;

                // Xử lý chiều cao an toàn từ dữ liệu "h" của ESP32
                if (data['h'] != null) {
                    currentHeight = double.tryParse(data['h'].toString()) ?? 0.0;
                }

                isLed1On = data['led1'] ?? false;
                isLed2On = data['led2'] ?? false;
                isConnected = true;
            });
        }
    } catch (e) { if(mounted) setState(() => isConnected = false); }
}

Future<void> toggleLed(int id) async {
    try { await http.get(Uri.parse('http://$espIp/led$id')); fetchData(); } catch
(e) {}
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text("MÀN HÌNH ĐIỀU KHIỂN", style: TextStyle(fontWeight:
FontWeight.bold, color: Colors.white)),
            backgroundColor: Colors.blueAccent,
            centerTitle: true,

```

```

        actions: [IconButton(onPressed: () =>
Navigator.pushReplacementNamed(context, '/'), icon: const Icon(Icons.logout,
color: Colors.white))],
),
body: SingleChildScrollView(
padding: const EdgeInsets.all(20),
child: Column(
children: [
// Ô TRẠNG THÁI KẾT NỐI
_buildConnectionStatus(),
const SizedBox(height: 20),

// THÔNG TIN GIÀNG VIÊN & SINH VIÊN
_buildInfoCard(),
const SizedBox(height: 20),

// Ô TỔNG SỐ VẬT
_buildStatCard("TỔNG SỐ VẬT", total.toString(), Colors.blueGrey,
Icons.inventory, isLarge: true),
const SizedBox(height: 15),

// Ô CHIỀU CAO HIỆN TẠI
Row(children: [
Expanded(child: _buildStatCard("CHIỀU CAO (mm)",
currentHeight.toStringAsFixed(1), Colors.orange, Icons.straighten)),
const SizedBox(width: 15),
Expanded(child: _buildStatCard("TRẠNG THÁI", currentHeight > 5 ?
"CÓ VẬT" : "TRỐNG", currentHeight > 5 ? Colors.teal : Colors.grey,
Icons.sensors)),
]),
const SizedBox(height: 15),

// Ô PHÂN LOẠI MÀU
Row(children: [
Expanded(child: _buildStatCard("VẬT ĐỎ", red.toString(),
Colors.redAccent, Icons.circle)),
const SizedBox(width: 15),
Expanded(child: _buildStatCard("VẬT XANH", blue.toString(),
Colors.blueAccent, Icons.circle_outlined)),
]),
const SizedBox(height: 30),
const Align(alignment: Alignment.centerLeft, child: Text("ĐIỀU KHIỂN
THIẾT BỊ", style: TextStyle(fontWeight: FontWeight.bold, color:
Colors.blueGrey))),
```

```
        const Divider(),  
  
        _buildLedSwitch("Đèn Soi Cảm Biến", isLed1On, 1, Icons.lightbulb,  
Colors.orange),  
        _buildLedSwitch("Đèn Cảnh Báo Hệ Thống", isLed2On, 2,  
Icons.notification_important, Colors.redAccent),  
    ],  
),  
,  
);  
}  
  
Widget _buildConnectionStatus() {  
    return Container(  
        padding: const EdgeInsets.all(12),  
        decoration: BoxDecoration(  
            color: isConnected ? Colors.green.withOpacity(0.1) :  
Colors.red.withOpacity(0.1),  
            borderRadius: BorderRadius.circular(15),  
            border: Border.all(color: isConnected ? Colors.green : Colors.red),  
        ),  
        child: Row(mainAxisAlignment: MainAxisAlignment.center, children: [  
            Icon(isConnected ? Icons.wifi : Icons.wifi_off, color: isConnected ?  
Colors.green : Colors.red, size: 20),  
            const SizedBox(width: 10),  
            Text(isConnected ? "KẾT NỐI: $espIp" : "MẤT KẾT NỐI ESP32",  
                style: TextStyle(fontWeight: FontWeight.bold, color: isConnected ?  
Colors.green : Colors.red)),  
        ]),  
    );  
}  
  
Widget _buildInfoCard() {  
    return Container(  
        width: double.infinity,  
        padding: const EdgeInsets.all(15),  
        decoration: BoxDecoration(color: Colors.white, borderRadius:  
BorderRadius.circular(20), boxShadow: const [BoxShadow(color: Colors.black12,  
blurRadius: 5)]),  
        child: Column(children: [  
            const Text("GVHD: Nguyễn Kim Suyên", style: TextStyle(fontWeight:  
FontWeight.bold, color: Colors.blueGrey)),  
            const Text("SVTH: Khoa - Bách", style: TextStyle(color: Colors.black54)),  
        ]),  
    );  
}
```

```

    }

    Widget _buildStatCard(String title, String value, Color color, IconData icon,
{bool isLarge = false}) {
    return Container(
        padding: EdgeInsets.all(isLarge ? 25 : 18),
        decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(25),
            boxShadow: const [BoxShadow(color: Colors.black12, blurRadius: 8, offset:
Offset(0, 4))])
    ),
    child: Column(children: [
        Icon(icon, color: color, size: isLarge ? 40 : 28),
        const SizedBox(height: 8),
        Text(title, style: const TextStyle(fontSize: 10, fontWeight:
FontWeight.bold, color: Colors.grey)),
        Text(value, style: TextStyle(color: color, fontSize: isLarge ? 38 : 24,
fontWeight: FontWeight.bold)),
    ]),
);
}

Widget _buildLedSwitch(String title, bool val, int id, IconData icon, Color
activeColor) {
    return Card(
        elevation: 1,
        margin: const EdgeInsets.symmetric(vertical: 6),
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),
        child: SwitchListTile(
            secondary: Icon(icon, color: val ? activeColor : Colors.grey),
            title: Text(title, style: const TextStyle(fontSize: 14, fontWeight:
FontWeight.bold)),
            value: val,
            onChanged: isConnected ? (v) => toggleLed(id) : null,
            activeColor: activeColor,
        ),
    );
}
}

```

PUBSPEC.YAML

```
dependencies:  
  flutter:  
    sdk: flutter  
  http: ^1.1.0  
  shared_preferences: ^2.2.2 # Thêm dòng này
```