

NAME: HAIDER NAWAZ

SAP ID: 55330

SUBJECT: APP DEVELOPMENT

**SMART\_HOME\_DASHBOARD:**

**MAIN.DART:**

```
import 'package:flutter/material.dart';

void main() {
  runApp(SmartHomeApp());
}

class SmartHomeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Smart Home Dashboard',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        cardColor: Colors.white,
        scaffoldBackgroundColor: Colors.grey[200],
      ),
      home: DashboardScreen(),
    );
  }
}

class DashboardScreen extends StatefulWidget {
  @override
  State<DashboardScreen> createState() => _DashboardScreenState();
}
```

## ASSIGNMENT-03

```
class _DashboardScreenState extends State<DashboardScreen> {  
  List<Map<String, dynamic>> devices = [  
    {  
      "name": "Living Room Light",  
      "type": "Light",  
      "room": "Living Room",  
      "isOn": false,  
      "brightness": 50,  
      "icon": Icons.lightbulb_outline  
    },  
    {  
      "name": "Bedroom Fan",  
      "type": "Fan",  
      "room": "Bedroom",  
      "isOn": true,  
      "brightness": 70,  
      "icon": Icons.toys  
    }  
  ];  
  
  void showAddDeviceDialog() {  
    TextEditingController nameCtrl = TextEditingController();  
    TextEditingController roomCtrl = TextEditingController();  
    String selectedType = "Light";  
    bool defaultStatus = false;  
  
    showDialog(  
      context: context,  
      builder: (context) {  
        return AlertDialog(  
          title: Text("Add New Device"),  
          content: Column(  
            mainAxisSize: MainAxisSize.min,
```

## ASSIGNMENT-03

```
children: [
    TextField(
        controller: nameCtrl,
        decoration: InputDecoration(labelText: "Device Name"),
    ),
    TextField(
        controller: roomCtrl,
        decoration: InputDecoration(labelText: "Room Name"),
    ),
    DropdownButton<String>(
        value: selectedType,
        items: ["Light", "Fan", "AC", "Camera"]
        .map((e) => DropdownMenuItem(
            child: Text(e),
            value: e,
        )))
        .toList(),
        onChanged: (val) {
            setState(() {
                selectedType = val!;
            });
        },
    ),
    SwitchListTile(
        value: defaultStatus,
        title: Text("Status (ON / OFF)"),
        onChanged: (val) {
            setState(() {
                defaultStatus = val;
            });
        },
    ),
],
```

## ASSIGNMENT-03

```
)  
],  
)  
actions: [  
    TextButton(  
        onPressed: () {  
            Navigator.pop(context);  
        },  
        child: Text("Cancel")),  
    ElevatedButton(  
        onPressed: () {  
            setState(() {  
                devices.add({  
                    "name": nameCtrl.text,  
                    "type": selectedType,  
                    "room": roomCtrl.text,  
                    "isOn": defaultStatus,  
                    "brightness": 50,  
                    "icon": selectedType == "Light"  
                        ? Icons.lightbulb_outline  
                        : selectedType == "Fan"  
                        ? Icons.toys  
                        : selectedType == "AC"  
                        ? Icons.ac_unit  
                        : Icons.camera_alt  
                });  
            });  
            Navigator.pop(context);  
        },  
        child: Text("Add"))  
    ],
```

## ASSIGNMENT-03

```
    );
},
);

}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Smart Home Dashboard"),
            leading: Icon(Icons.menu),
            actions: [
                Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: CircleAvatar(backgroundColor: Colors.blue),
                )
            ],
        ),
        body: Padding(
            padding: const EdgeInsets.all(12.0),
            child: GridView.builder(
                itemCount: devices.length,
                gridDelegate:
                    SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2),
                itemBuilder: (context, index) {
                    var device = devices[index];
                    return InkWell(
                        onTap: () {
                            Navigator.push(

```

## ASSIGNMENT-03

```
context,  
MaterialPageRoute(  
    builder: (_) => DeviceDetailsScreen(  
        device: device,  
        onUpdate: () {  
            setState(() {});  
        },  
    ),  
,  
);  
,  
child: Card(  
    elevation: 3,  
    child: Padding(  
        padding: const EdgeInsets.all(8.0),  
        child: Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: [  
                Icon(device["icon"], size: 40),  
                SizedBox(height: 10),  
                Text(device["name"],  
                    textAlign: TextAlign.center,  
                    style: TextStyle(fontWeight: FontWeight.bold)),  
                Switch(  
                    value: device["isOn"],  
                    onChanged: (val) {  
                        setState(() {  
                            device["isOn"] = val;  
                        });  
                    },  
                ),  
            ],  
        ),  
    ),  
);
```

ASSIGNMENT-03

```
        Text(device["isOn"] ? "Status: ON" : "Status: OFF")  
    ],  
    ),  
    ),  
    ),  
);  
,  
,  
),  
),  
floatingActionButton: FloatingActionButton(  
    onPressed: showAddDeviceDialog,  
    child: Icon(Icons.add),  
,  
);  
}  
}  
  
class DeviceDetailsScreen extends StatefulWidget {  
    final Map<String, dynamic> device;  
    final VoidCallback onUpdate;  
    DeviceDetailsScreen({required this.device, required this.onUpdate});  
  
    @override  
    State<DeviceDetailsScreen> createState() => _DeviceDetailsScreenState();  
}  
  
  
class _DeviceDetailsScreenState extends State<DeviceDetailsScreen> {  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(  
                title: Text(widget.device["name"]),  
            ),  
            body: Container(  
                padding: EdgeInsets.all(16),  
                child: Column(  
                    mainAxisAlignment: MainAxisAlignment.start,  
                    crossAxisAlignment: CrossAxisAlignment.start,  
                    children: [  
                        Text("Name: ${device["name"]}"),  
                        Text("Type: ${device["type"]}"),  
                        Text("Status: ${device["isOn"] ? "ON" : "OFF"}"),  
                        Text("IP Address: ${device["ipAddress"]}"),  
                        Text("Last Seen: ${device["lastSeen"]}"),  
                        Text("Battery Level: ${device["batteryLevel"]} %"),  
                        Text("Signal Strength: ${device["signalStrength"]} dBm"),  
                        Text("Temperature: ${device["temperature"]} °C"),  
                        Text("Humidity: ${device["humidity"]} %"),  
                        Text("Light Level: ${device["lightLevel"]} lux"),  
                        Text("Motion Detected: ${device["motionDetected"]}"),  
                        Text("Water Leaking: ${device["waterLeaking"]}"),  
                        Text("Door Opened: ${device["doorOpened"]}"),  
                        Text("Window Opened: ${device["windowOpened"]}"),  
                        Text("Smoke Detected: ${device["smokeDetected"]}"),  
                        Text("Carbon Monoxide Detected: ${device["carbonMonoxideDetected"]}"),  
                        Text("Vibration Detected: ${device["vibrationDetected"]}"),  
                        Text("Water Level: ${device["waterLevel"]} mm"),  
                        Text("Pump Status: ${device["pumpStatus"]}"),  
                        Text("Fan Status: ${device["fanStatus"]}"),  
                        Text("Heater Status: ${device["heaterStatus"]}"),  
                        Text("Light Status: ${device["lightStatus"]}"),  
                        Text("Lock Status: ${device["lockStatus"]}"),  
                        Text("Door Lock Status: ${device["doorLockStatus"]}"),  
                        Text("Window Lock Status: ${device["windowLockStatus"]}"),  
                        Text("Smoke Detector Status: ${device["smokeDetectorStatus"]}"),  
                        Text("Carbon Monoxide Detector Status: ${device["carbonMonoxideDetectorStatus"]}"),  
                        Text("Vibration Sensor Status: ${device["vibrationSensorStatus"]}"),  
                        Text("Water Sensor Status: ${device["waterSensorStatus"]}"),  
                        Text("Pump Control: ${device["pumpControl"]}"),  
                        Text("Fan Control: ${device["fanControl"]}"),  
                        Text("Heater Control: ${device["heaterControl"]}"),  
                        Text("Light Control: ${device["lightControl"]}"),  
                        Text("Lock Control: ${device["lockControl"]}"),  
                        Text("Door Lock Control: ${device["doorLockControl"]}"),  
                        Text("Window Lock Control: ${device["windowLockControl"]}"),  
                        Text("Smoke Detector Control: ${device["smokeDetectorControl"]}"),  
                        Text("Carbon Monoxide Detector Control: ${device["carbonMonoxideDetectorControl"]}"),  
                        Text("Vibration Sensor Control: ${device["vibrationSensorControl"]}"),  
                        Text("Water Sensor Control: ${device["waterSensorControl"]}"),  
                        Text("Pump Status: ${device["pumpStatus"]}"),  
                        Text("Fan Status: ${device["fanStatus"]}"),  
                        Text("Heater Status: ${device["heaterStatus"]}"),  
                        Text("Light Status: ${device["lightStatus"]}"),  
                        Text("Lock Status: ${device["lockStatus"]}"),  
                        Text("Door Lock Status: ${device["doorLockStatus"]}"),  
                        Text("Window Lock Status: ${device["windowLockStatus"]}"),  
                        Text("Smoke Detector Status: ${device["smokeDetectorStatus"]}"),  
                        Text("Carbon Monoxide Detector Status: ${device["carbonMonoxideDetectorStatus"]}"),  
                        Text("Vibration Sensor Status: ${device["vibrationSensorStatus"]}"),  
                        Text("Water Sensor Status: ${device["waterSensorStatus"]}"),  
                        Text("Pump Control: ${device["pumpControl"]}"),  
                        Text("Fan Control: ${device["fanControl"]}"),  
                        Text("Heater Control: ${device["heaterControl"]}"),  
                        Text("Light Control: ${device["lightControl"]}"),  
                        Text("Lock Control: ${device["lockControl"]}"),  
                        Text("Door Lock Control: ${device["doorLockControl"]}"),  
                        Text("Window Lock Control: ${device["windowLockControl"]}"),  
                        Text("Smoke Detector Control: ${device["smokeDetectorControl"]}"),  
                        Text("Carbon Monoxide Detector Control: ${device["carbonMonoxideDetectorControl"]}"),  
                        Text("Vibration Sensor Control: ${device["vibrationSensorControl"]}"),  
                        Text("Water Sensor Control: ${device["waterSensorControl"]}"),  
                    ],  
                ),  
            ),  
        );  
    }  
}
```

ASSIGNMENT-03

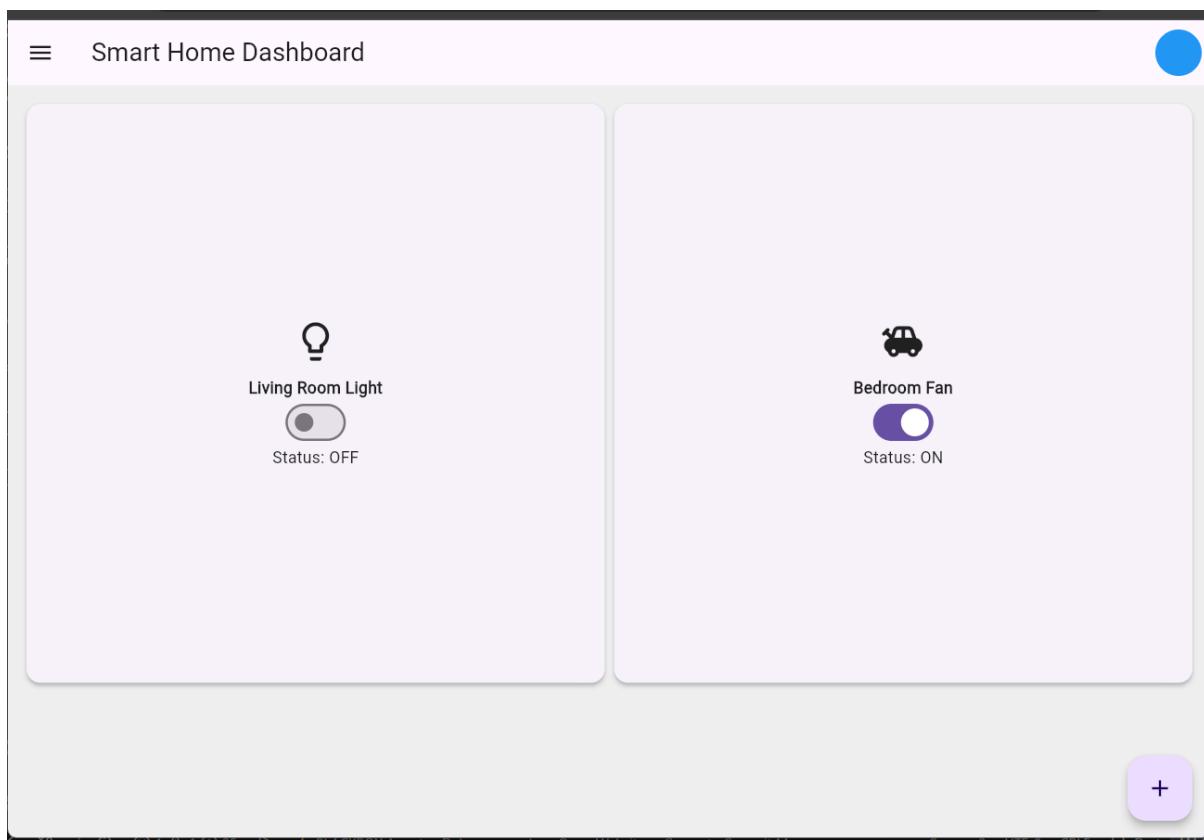
```
),
body: Padding(
padding: const EdgeInsets.all(20.0),
child: Column(
children: [
Icon(widget.device["icon"], size: 120),
SizedBox(height: 20),
Text(
widget.device["isOn"] ? "Device is ON" : "Device is OFF",
style: TextStyle(fontSize: 18),
),
SizedBox(height: 20),

// Slider for brightness or speed
Text("Control Level: ${widget.device["brightness"]}"),
Slider(
min: 0,
max: 100,
value: widget.device["brightness"].toDouble(),
onChanged: (val) {
setState(() {
widget.device["brightness"] = val.toInt();
});
widget.onUpdate();
},
),
SwitchListTile(
title: Text("Turn ON / OFF"),
value: widget.device["isOn"],
onChanged: (val) {
setState(() {
```

## ASSIGNMENT-03

```
        widget.device["isOn"] = val;  
    });  
    widget.onUpdate();  
},  
,  
,  
,  
);  
}  
}
```

## RUNNING SCREENS



### ASSIGNMENT-03

