Advanced IoT Course

Module 1: Advanced IoT Architectures and Systems

- **1.1 Complex IoT System Design**
- **Objective**: Understand the architecture of large-scale IoT systems, including device communication, data storage, and user interfaces.
- **Topics**:
  - IoT Device Layer: Sensors, actuators, microcontrollers (ESP32, Raspberry Pi).
  - Connectivity Layer: Wi-Fi, MQTT, LoRa, Zigbee, NB-IoT.
  - Data Processing Layer: Edge computing, cloud computing, hybrid architectures.
  - Application Layer: Data visualization, analytics, integration with apps, dashboards.

- **1.2 Designing Scalable IoT Systems**
- **Objective**: Learn how to design IoT systems that scale efficiently.
- **Topics**:
  - Device management at scale.
  - Data flow optimization.
  - Redundancy and fault tolerance.
  - Handling millions of devices and high-volume data streams.

Module 2: IoT Networking Protocols and Security

- **2.1 Advanced Networking Protocols**
- **Objective**: Study protocols beyond MQTT and HTTP for large-scale IoT applications.
- **Topics**:
  - **CoAP** (Constrained Application Protocol) for low-power devices.
  - **AMQP** (Advanced Message Queuing Protocol) for enterprise-scale communication.
  - **LoRaWAN** for long-range, low-power IoT networks.
  - **5G and NB-IoT** for next-gen connectivity.

- **2.2 IoT Security Challenges and Solutions**
- **Objective**: Master the security risks in IoT and how to mitigate them.
- **Topics**:
  - Data encryption: Transport Layer Security (TLS), Public Key Infrastructure (PKI).
  - Device authentication: Certificate-based authentication, OAuth, and JWT.
  - Attack prevention: DDoS attacks, man-in-the-middle (MITM) attacks, and eavesdropping.
  - Secure firmware updates.
- **Tools/Techniques:**
- **Wireshark** for packet sniffing and security testing.
- **OpenSSL** for generating certificates and securing communications.

Module 3: Edge Computing for IoT

- **3.1 Introduction to Edge Computing**
- **Objective**: Learn about edge computing and why it is essential for time-sensitive IoT applications.
- **Topics**:
  - Edge vs. Cloud computing: Benefits and trade-offs.
  - Latency reduction: Processing data closer to the source.
  - Real-time decision-making on the device.
  - Edge devices: Raspberry Pi, Nvidia Jetson, Intel NUC, and ESP32.

3.2 Implementing Edge Computing

- **Objective**: Implement edge computing in an IoT system to handle data locally.
- **Projects**:
  - **Smart Surveillance System**: Use edge computing to analyze video feeds from cameras using a Raspberry Pi or Nvidia Jetson, without sending the data to the cloud.
  - **Predictive Maintenance**: Use sensors to monitor machinery, and apply edge analytics to predict failures in real time.

- **Code Example**: Edge Processing on ESP32

```
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";
const char* mqtt_server = "YourMQTTBroker";

WiFiClient espClient;
PubSubClient client(espClient);

#define SENSOR_PIN 34

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
  }
  client.setServer(mqtt_server, 1883);
}

void loop() {
  int sensorValue = analogRead(SENSOR_PIN);

  if (sensorValue > 500) {
    // Process locally and send only important data
    client.publish("iot/sensors/alert", "High sensor value detected");
  }

  delay(1000);
  client.loop();
}
```

Module 4: Cloud Integration for IoT

- **4.1 Cloud Platforms and IoT**
- **Objective**: Learn how to use cloud platforms to store, analyze, and visualize IoT data.
- **Topics**:
  - Overview of cloud platforms: **AWS IoT**, **Microsoft Azure IoT**, **Google Cloud IoT**, **IBM Watson**.
  - Data storage solutions: Relational and NoSQL databases (e.g., AWS DynamoDB, Google BigQuery).
  - Data processing pipelines: Use cloud services like AWS Lambda, Azure Functions.
  - Visualization tools: **Power BI**, **Grafana**, **Tableau**.

4.2 Cloud-to-Edge Integration

- **Objective**: Connect edge devices to cloud platforms and enable hybrid IoT systems.
- **Topics**:
  - **AWS Greengrass**: Bring cloud capabilities to the edge.
  - **Azure IoT Edge**: Extend cloud intelligence to IoT devices.
  - Hybrid architecture: Process some data on the edge and the rest in the cloud.
  - Real-time data streaming: Use **AWS Kinesis**, **Azure Event Hub**, or **Google Cloud Pub/Sub** for real-time data ingestion.

- **Code Example**: Sending Data to AWS IoT Core and AWS Lambda

```
#include <WiFi.h>

#include <PubSubClient.h>


const char* ssid = "YourWiFiSSID";

const char* password = "YourWiFiPassword";

const char* mqtt_server = "YourAWSIoTEndpoint";

const char* topic = "iot/sensors/data";

WiFiClient espClient;

PubSubClient client(espClient);


void setup() {

 Serial.begin(115200);

 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED) {

   delay(1000);

 }

 client.setServer(mqtt_server, 8883);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Simulate sending sensor data to AWS IoT Core
  String payload = "{\"temperature\": 24.5}";
  client.publish(topic, payload.c_str());
  delay(5000);
}

void reconnect() {
  while (!client.connected()) {
    if (client.connect("ESP32Client")) {
      // Successfully connected
    } else {
      delay(5000);
    }
  }
}
```

Module 5: Machine Learning for IoT

- **5.1 Introduction to Machine Learning in IoT**
- **Objective**: Apply machine learning models to IoT data to make intelligent decisions and predictions.
- **Topics**:
  - Basic ML concepts: Classification, regression, and clustering.
  - Training models: Using historical IoT data to train models.
  - Model deployment: Deploying models on edge devices (using TensorFlow Lite or TinyML) or in the cloud

5.2 Real-Time Predictive Analytics

- **Objective**: Learn how to integrate machine learning into IoT systems for real-time data analysis and predictions.
- **Projects**:
  - **Predictive Maintenance**: Use sensor data to predict when a machine or asset is likely to fail.
  - **Smart Agriculture**: Use weather and soil data to predict crop yields and irrigation needs.

Example: Predictive Maintenance Using IoT Data

- **Sensors**: Vibration, temperature, and humidity sensors.
- **Model**: Train a decision tree or logistic regression model on historical sensor data to predict failures.
- **Deployment**: Use edge devices like Raspberry Pi or ESP32 to process the sensor data and make predictions locally.

Module 6: Advanced IoT Applications

- **6.1 IoT in Smart Cities**
- **Objective**: Learn how IoT can be used in smart city applications for traffic management, pollution monitoring, waste management, and more.
- **Topics**:
  - Smart traffic lights: Using real-time traffic data to control signals.
  - Pollution monitoring: Using environmental sensors to measure air quality.
  - Waste management: Using sensors in garbage bins to optimize waste collection.

- **6.2 Industrial IoT (IIoT)**
- **Objective**: Explore how IoT is revolutionizing industries like manufacturing, logistics, and energy.
- **Topics**:
  - Real-time monitoring of machines and equipment.
  - Predictive analytics for machine performance.
  - Use of IoT for supply chain optimization.