UNIT IV PROTOTYPING AND DESIGNING SOFTWARE FOR IOT APPLICATIONS

Title: Internet and Web/Cloud services software development.

In IoT applications, once data is collected from devices and transmitted through gateways, it needs to be processed, visualized, and stored on the Internet or cloud-based platforms. Cloud services enable real-time monitoring, analytics, remote control, and automation features that make IoT systems scalable and intelligent. Developing software for these platforms involves creating APIs, data pipelines, dashboards, storage, and security mechanisms tailored for IoT devices and user interfaces.



Why Use Web/Cloud Services in IoT

- Remote Accessibility: Monitor and control devices from anywhere.
- Data Storage and Processing: Store large volumes of sensor data over time.
- Scalability: Easily expand to accommodate more devices and users.
- Real-Time Analytics: Make smart decisions based on live data.
- Integration with AI/ML: Enable predictive analytics and automation.
- Maintenance and Updates: Update firmware and monitor device status remotely.

Cloud Architecture for IoT

A typical cloud-based IoT architecture consists of:

- 1. Device Layer Sensors and actuators that generate data.
- 2. Gateway/Edge Layer Preprocessing and protocol translation.
- 3. Network Layer Communication (Wi-Fi, 4G, LoRaWAN, etc.)

- 4. Cloud Platform Layer Application logic, storage, analytics.
- 5. Application/User Layer Dashboards, alerts, and mobile/web apps.



Popular IoT Cloud Platforms

- AWS IoT Core Secure device communication, data processing, rules engine, and cloud storage.
- Microsoft Azure IoT Hub Bi-directional communication, device provisioning, and integrations with Azure services.
- Google Cloud IoT Device manager, Pub/Sub messaging, and BigQuery analytics.
- ThingSpeak Open-source, ideal for academic and small-scale projects with real-time charts.
- Blynk Rapid app development for mobile IoT interfaces.



Web and Cloud Development Components

1. Data Ingestion

- Use protocols like MQTT, HTTP REST API, WebSocket.
- Devices publish data to cloud services using unique topics and keys.
- 2. Backend Development
 - APIs built using platforms like Node.js, Python (Flask/FastAPI), Java (Spring Boot).
 - Responsible for receiving sensor data, storing it in databases, and serving data to frontend or dashboards.
- 3. Database Storage
 - SQL (MySQL, PostgreSQL) Structured data (device logs, users).
 - NoSQL (MongoDB, InfluxDB) Best for time-series sensor data.
- 4. Data Visualization and UI
 - Dashboards built with HTML/CSS/JavaScript, or frameworks like React.js, Vue.js.
 - Libraries: Chart.js, D3.js, Google Charts for graphing data.
- 5. User Authentication & Security
 - Secure user login using OAuth2 or JWT.
 - HTTPS encryption, token validation, and access control policies.

Example: Simple IoT Weather Monitoring with Cloud

Setup:

- Device: Arduino + DHT11 (temperature/humidity)
- Connectivity: ESP8266 or ESP32 module
- Platform: ThingSpeak

Workflow:

- 1. Sensor reads temperature every 10 seconds.
- 2. ESP8266 connects to Wi-Fi and sends data via HTTP POST to ThingSpeak.
- 3. Data is stored and visualized on ThingSpeak's real-time dashboard.
- 4. Alerts (email/SMS) configured when thresholds are breached.