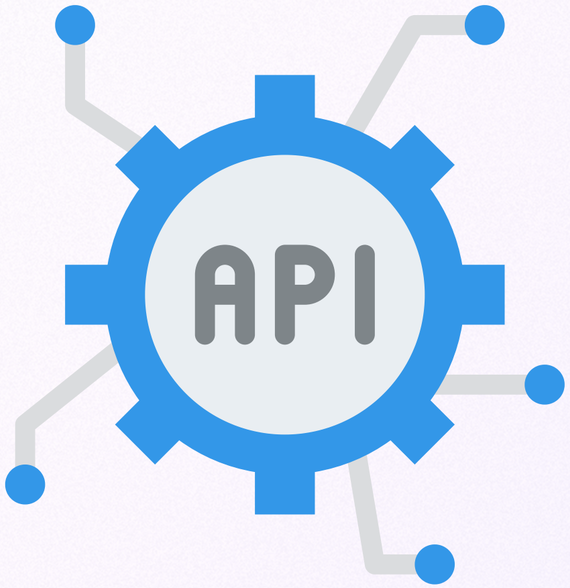
API Creation for IoT Platform

Development Process, Technology Stack, and Demonstration

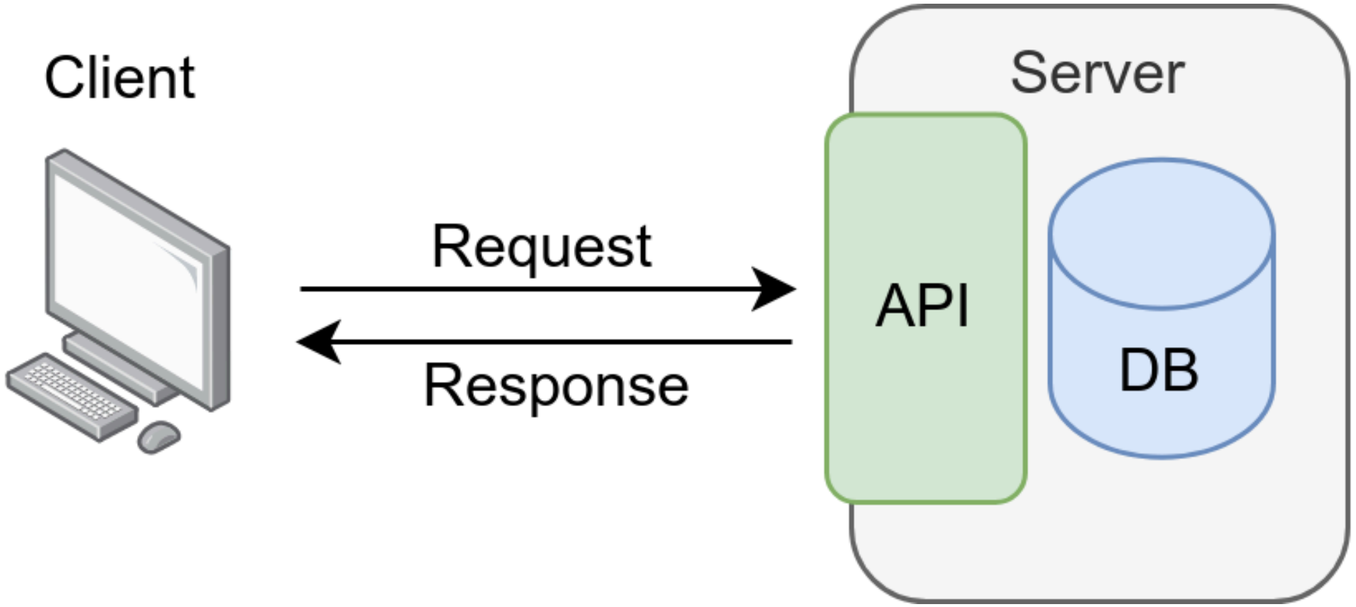


Ram Chandra Giri
Presentor

TABLE OF CONTENTS

WHAT IS API

APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.



Development Process

- **Planning:** Defined objectives for MQTT-based IoT platform and backend API requirements.
- **Design:** Created API architecture to handle device management, data storage, and visualization.
- **Implementation:** Developed APIs using FastAPI for efficiency and scalability.
- **Testing:** Conducted unit and integration tests for reliability.
- **Deployment:** Deployed using Docker for containerization and ensured scalability.

Development Process

- **Planning:** Defined objectives for MQTT-based IoT platform and backend API requirements.
- **Design:** Created API architecture to handle device management, data storage, and visualization.
- **Implementation:** Developed APIs using FastAPI for efficiency and scalability.
- **Testing:** Conducted unit and integration tests for reliability.
- **Deployment:** Deployed using Docker for containerization and ensured scalability.

Technology Stack Summary

Component	Technology Used
Programming Language	Python
Backend Framework	FastAPI
Frontend Framework	Streamlit, React
Database	MongoDB
MQTT Broker	Mosquitto
Authentication	JWT for secure token-based authentication
Data Visualization	Recharts, Plotly

Core Features:

1. **Device Management:**
   - Add, update, or delete IoT devices
   - Assign devices to users and roles
2. **Data Storage:**
   - Store incoming MQTT data in MongoDB/InfluxDB
   - Enable efficient time-series queries
3. **Visualization:**
   - Fetch processed data for charting and analysis on the frontend
4. **User Authentication:**
   - Secure endpoints with JWT tokens

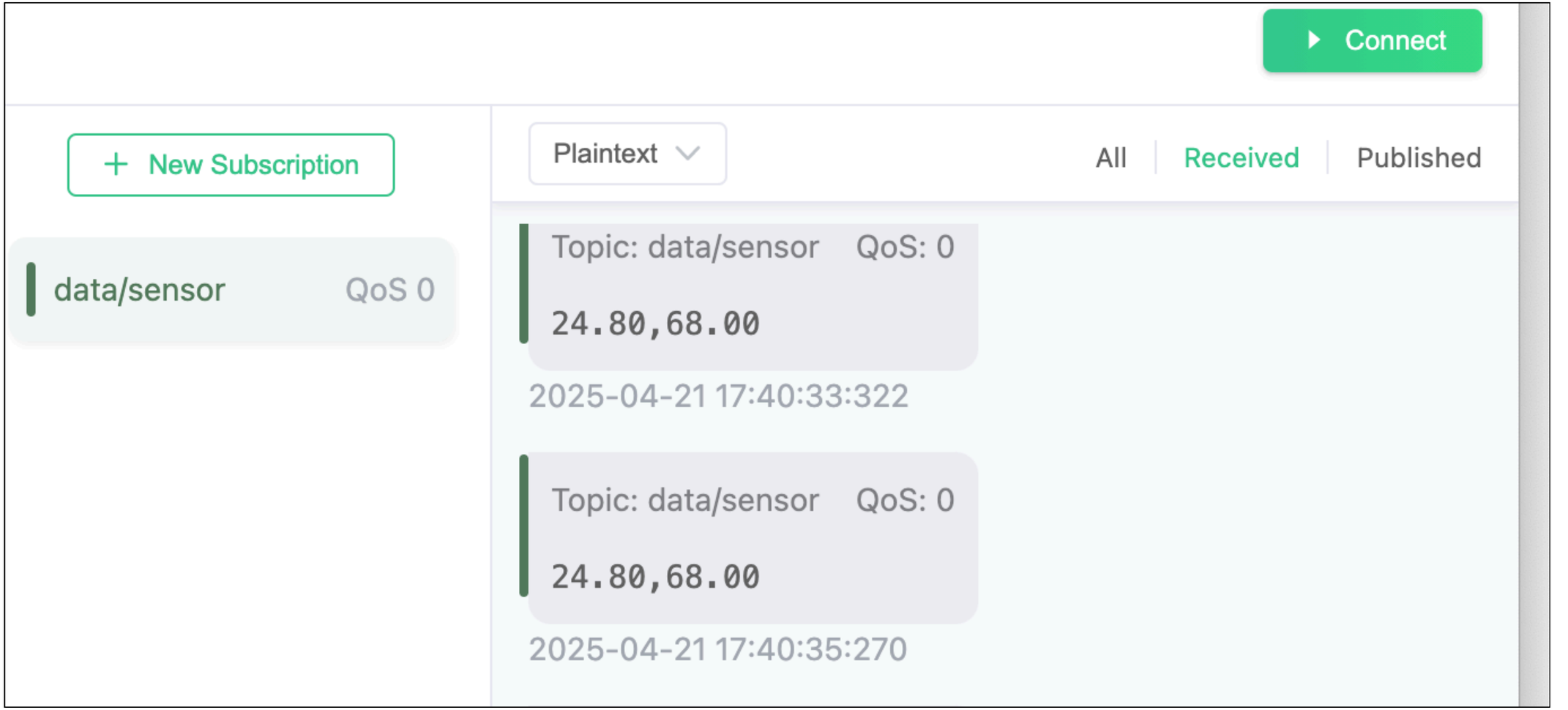
How it works?

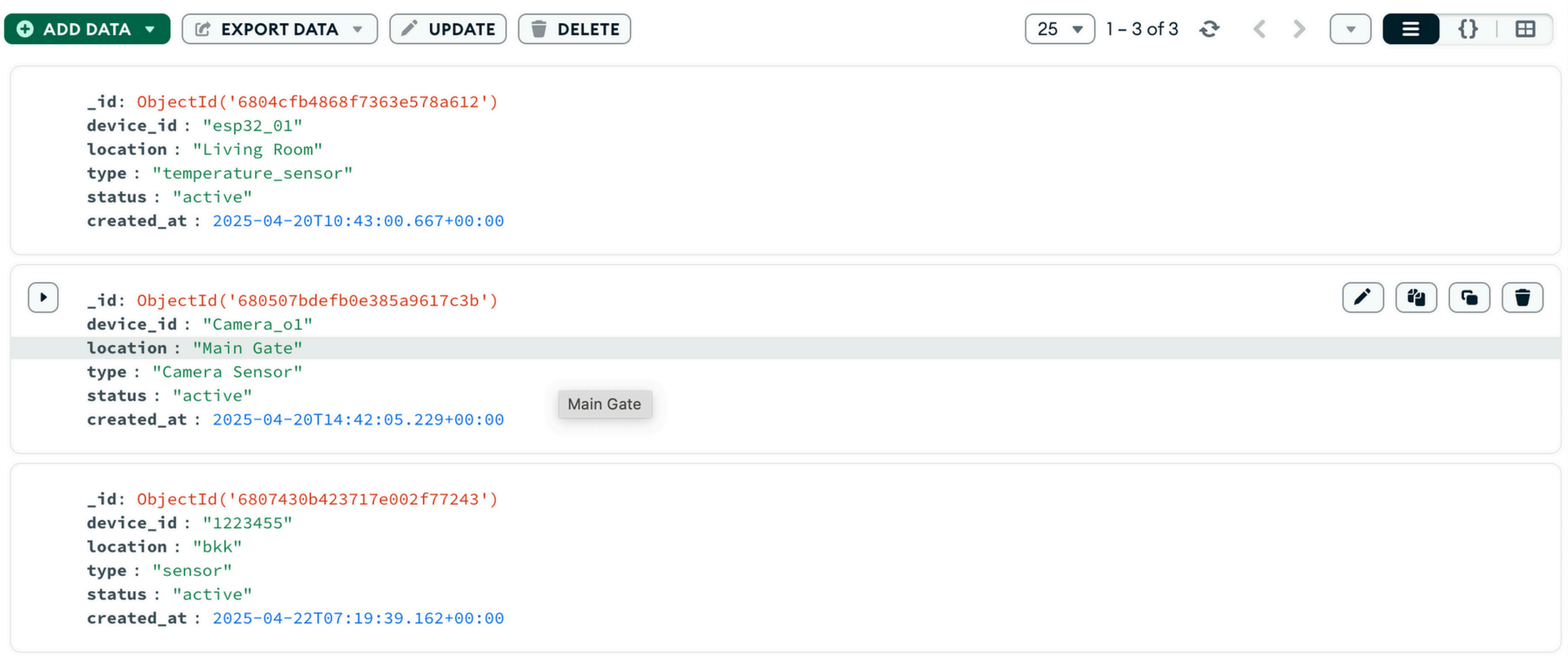
Device Data Flow:
Devices publish data to the MQTT broker
MQTT broker forwards data to FastAPI for processing

Data Storage Process:
FastAPI receives and validates the data
Stores structured data in MongoDB and time-series data in InfluxDB

Data Retrieval for Visualization:
User requests processed data via FastAPI endpoints
API aggregates and serves data for frontend charts

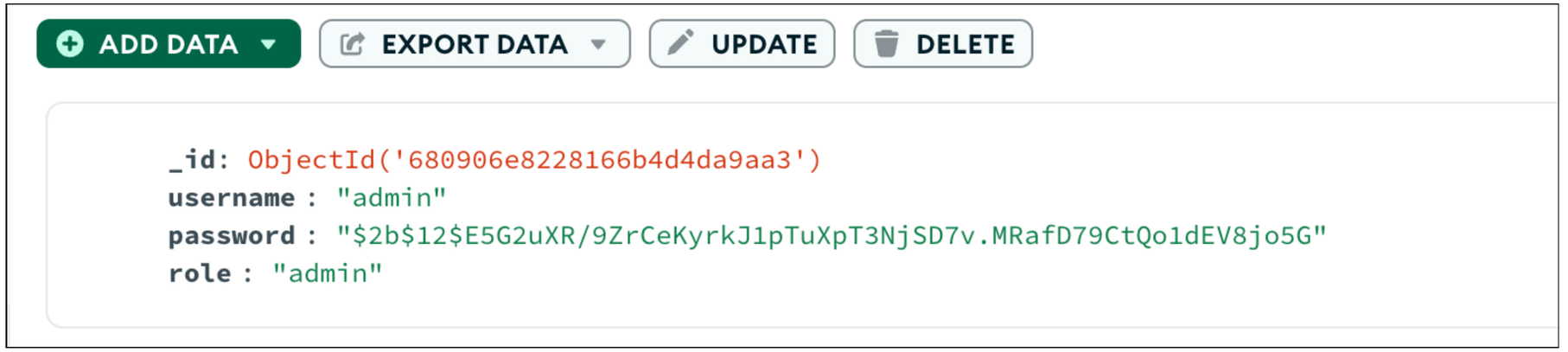
Secure Operations:
Role-based access ensures only authorized users access sensitive endpoints


Connect
New Subscription
Plaintext
All
Received
Published
data/sensor
QoS 0
Topic: data/sensor QoS: 0
24.80,68.00
2025-04-21 17:40:33:322
Topic: data/sensor QoS: 0
24.80,68.00
2025-04-21 17:40:35:270

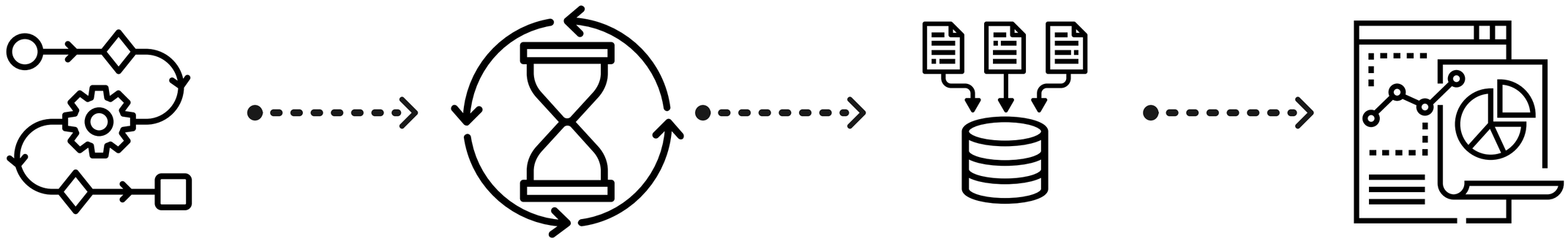

ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 – 3 of 3
_id: ObjectId('6804cfb4868f7363e578a612')
device_id : "esp32_01"
location : "Living Room"
type : "temperature_sensor"
status : "active"
created_at : 2025-04-20T10:43:00.667+00:00
_id: ObjectId('680507bdefb0e385a9617c3b')
device_id : "Camera_o1"
location : "Main Gate"
type : "Camera Sensor"
status : "active"
created_at : 2025-04-20T14:42:05.229+00:00
Main Gate
_id: ObjectId('6807430b423717e002f77243')
device_id : "1223455"
location : "bkk"
type : "sensor"
status : "active"
created_at : 2025-04-22T07:19:39.162+00:00

```python
def initialize_admin():
    admin_username = "admin"
    admin_password = "admin123"
    hashed_password = bcrypt.hash(admin_password)


    if not db["users"].find_one({"username": admin_username}):
        db["users"].insert_one({
            "username": admin_username,
            "password": hashed_password,
            "role": "admin"
        })
```



EXAMPLE WORK FLOW



1 - API Request

Client sends a request to the /add_device endpoint with device details.

2 - Processing

FastAPI validates the request, updates the database, and confirms the operation.

3 - Real Time Data

Devices publish live data via MQTT, stored via API to databases.

4 -Visualization

Frontend fetches real-time data through APIs to generate dashboards.

Conclusion

- API is the backbone of the IoT platform, enabling device interaction and data handling.
- Scalable architecture ensures future-proofing for additional features.
- Demonstrated a robust integration of MQTT, FastAPI, and databases.

THANK YOU