Usecase

Indexing various services in IoT platforms

Aim :- To design and implement a data storage mechanism for an IoT platform that can efficiently store, retrieve, and index data from a wide range of heterogeneous devices The mechanism should support fast querying and flexible schema to handle new device types easily.

Explanation :- A generic IoT platform collects data from different types of devices such as temperature sensors, humidity, sensors, and energy meters. Each device generates data in various formats and at different intervals. Relational database would require a fixed schema making it difficult to handle new or variable device data structures.

To overcome this issue, a JSON-based documents oriented database like mango DB is preferred. Mango DB stores data in flexible JSON-like documents, allowing each device type to store its own data attributes. The makes the system adaptable and scalable.

Why Mango DB ?

→ Schema-less structure supports new device types dynamically

→ Documents are stored in JSON format suitable for IOT data.

→ Supports indexing on single and multiple fields.

→ Allows fast retrievals of data for specific devices or parameters.

→ Scalable and efficient for handling large volumes real-time data.

Example JSON Documents :-

Temperature sensor Example:

```
{
"device_id": "D1001",
"device_type": "Temperature sensor",
"location_id": "100",
"timeStamp": "2025-11-03T10:45:00+",
"data": {
"temperature": 2805,
"Humidity": 70
},
"status": "active"
}
```

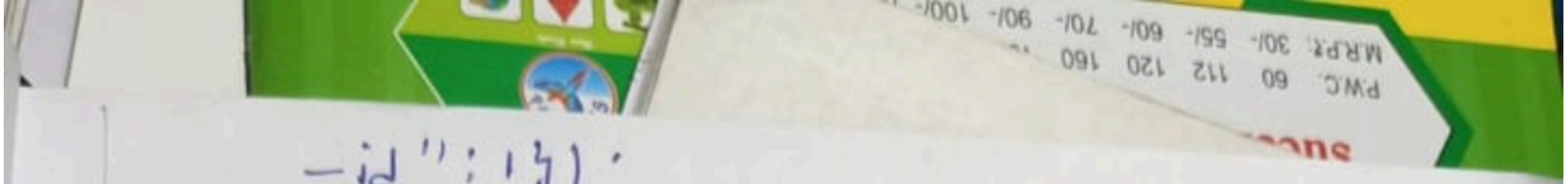
Energy meter Example:

```
{
"device_id": "E 2001",
"device_type": Energy meter",
"location_id": "1001",
"timeStamp": "2025-1103 T10:48:007",
"data": {
"voltage": 220,
"current": 1.5,
"power": 330
},
"status": "active"
}
```

Index creation in MongoDB :-

" Create single and compound indexes

```
db.iot_devices.Create index({"device_id":1});
db.iot_devices createindex({"device_id":1, "loca
```



-id": 1});

db.iot-devices create index({"data temperature":1});

Sample Queries:-

1. Retrieve all recorder of a particular device:
   db-iot-devices find({"device-id":"1)1001"});
2. Retrieve data for a device in a particular location;
   db.iot-devices find({"device-id":1,"location-id"});
3. Retrieve temperature reading from all devices
   db.iot-devices find({"data.temperature":
   {(exists:true}});

Result:- Using MongoDB allows the platform to store and retrieve data from devices without modifying the database structure indexes helps in fast query execution especially when searching data by device location or parameters the temperature. The System remains scalable and adaptable to new device types introduced in the future