Introduction to MongoDB

1. What is MongoDB?

- A **NoSQL**, **document-oriented database**.
- Stores data as **BSON (Binary JSON)**.
- Flexible schema (no need for fixed table structures).

2. Key Features

- Schema-less design
- High performance and scalability
- Supports **CRUD** operations
- Easy integration with Node.js using **Mongoose**

3. Installation

- Install MongoDB from mongodb.com
- Run MongoDB service:
- mongod
- Start shell:
- mongo

4. Basic Commands

```
show dbs            # list databases
use myDatabase      # switch/create db
db.createCollection("users")
show collections
```

⚙️ **Page 2: CRUD Operations**

1. Insert Documents

```
db.users.insertOne({ name: "John", age: 25 })
db.users.insertMany([{ name: "Asha" }, { name: "Kiran" }])
```

2. Read Documents

```
db.users.find()
```

db.users.find({ age: { $gt: 20 } })

db.users.findOne({ name: "John" })

3. Update Documents

db.users.updateOne({ name: "John" }, { $set: { age: 26 } })

db.users.updateMany({}, { $set: { status: "active" } })

4. Delete Documents

db.users.deleteOne({ name: "Asha" })

db.users.deleteMany({ age: { $lt: 18 } })

5. Query Operators

- $gt, $lt, $eq, $and, $or, $in
  Example:

db.users.find({ $or: [{ age: { $gt: 20 } }, { name: "Kiran" }] })

💾 **Page 3: Mongoose & Best Practices**

1. Connecting MongoDB with Node.js

```
const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/mydb')
  .then(() => console.log('Connected'))
  .catch(err => console.log(err));
```

2. Defining a Schema

```
const userSchema = new mongoose.Schema({
  name: String,
  age: Number,
  email: String
});

const User = mongoose.model('User', userSchema);
```

3. CRUD Example

```
// Create
```

```
await User.create({ name: "Meera", age: 22 });


// Read
const users = await User.find();


// Update
await User.updateOne({ name: "Meera" }, { age: 23 });


// Delete
await User.deleteOne({ name: "Meera" });
```

4. Best Practices

- Use **.env** for sensitive data (DB URI).
- Validate schema fields.
- Use indexes for faster queries.
- Handle connection errors with try-catch.