Experiment No. 10

TITLE
Create a MongoDB Document and Implement CRUD Operations on It.

PROBLEM STATEMENT
Design a MongoDB document schema and perform Create, Read, Update, and Delete (CRUD) operations on it using MongoDB Shell or Compass. The document will represent employee data and allow manipulation of the stored information through queries.

OBJECTIVE

- Understand document structure in MongoDB (NoSQL).
- Create a MongoDB database and collection.
- Insert one or more documents into the collection.
- Perform CRUD operations: Insert (Create), Find (Read), Update, Delete.
- Explore MongoDB Shell or Compass for interacting with the database.

PLATFORM REQUIRED

- Operating System: Windows or Linux
- Software/Tools: MongoDB, MongoDB Compass (GUI), MongoDB Shell or Node.js (optional)

THEORY
MongoDB is a NoSQL database that stores data in a document-oriented format called BSON (Binary JSON). Each document is a key-value pair structure and is stored in collections.

CRUD refers to the four basic operations that can be performed on a document in the database:

Operation	Description
Create	Insert documents into a collection
Read	Retrieve documents from the collection
Update	Modify existing documents
Delete	Remove documents from the collection

STEP BY STEP ALGORITHM

1. Start the MongoDB service.
2. Connect to MongoDB using Mongo Shell or Compass.

3. Create a new database company.
4. Create a collection employees.
5. Insert sample employee documents.
6. Retrieve (read) all or specific documents using queries.
7. Update selected fields in a document.
8. Delete one or more documents.
9. Optionally, use filtering, projections, and sorting.

MONGODB COMMANDS

- **Start Mongo Shell**

mongosh

- **Create Database**

use company

- **Create & Insert Document into Collection**

```
db.employees.insertOne({
  emp_id: 101,
  name: "Alice",
  department: "HR",
  salary: 50000
})

db.employees.insertMany([
  { emp_id: 102, name: "Bob", department: "IT", salary: 60000 },
  { emp_id: 103, name: "Charlie", department: "Finance", salary: 55000 }
])
```

- **Read (Find) Data**

```
db.employees.find()
db.employees.find({ department: "IT" })
db.employees.find({}, { name: 1, salary: 1 })
```

- **Update Document**

```
db.employees.updateOne(
  { emp_id: 103 },
  { $set: { salary: 58000 } }
)

db.employees.updateMany(
  { department: "IT" },
  { $inc: { salary: 2000 } }
)
```

- **Delete Document**

```
db.employees.deleteOne({ emp_id: 101 })
db.employees.deleteMany({ department: "HR" })
```

QUESTIONS

1. What is the difference between SQL and NoSQL databases?
2. Explain the document model of MongoDB.
3. How does MongoDB store data internally?
4. What is the difference between insertOne and insertMany?
5. How do you perform conditional updates in MongoDB?
6. Explain the usage of $set and $inc operators.
7. What happens if you update a field that does not exist?

CONCLUSION

This assignment introduces students to NoSQL database concepts using MongoDB. It covers how documents are created, accessed, modified, and deleted within a collection. Through this practical, students learn the flexibility and power of document-based storage and gain experience using CRUD operations in real-world database applications.