Lab Program3:

1. Create a pharmacy database containing **customers**, **medicines**, **suppliers**, **orders**, and **pharmacists**. Schema validation will enforce constraints like: age must be ≥ 18 for customers Expiry Date must be a valid ISO date in medicines Stock must be ≥ 0 Insert ≥10 documents into each collection and perform multi-collection aggregations.

a. Switch to database
b. Create collections with validation
c. Insert sample data (10 docs each)
d. Aggregation queries
   i. List all orders with customer name and medicine name
   ii. Total sales per medicine
   iii. Supplier-wise revenue
   iv. Medicines below stock threshold (e.g., < 70 units)
   v. Customer order summary

a. Switch to database

use pharmacyDB

b. Create collections with validation

```
db.createCollection("customers", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["custId", "name", "age", "phone"],
      properties: {
        custId: { bsonType: "int" },
        name: { bsonType: "string" },
        age: { bsonType: "int", minimum: 18 },
        phone: { bsonType: "string" }
      }
    }
  }
})

db.createCollection("medicines", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["medId", "medName", "supplierId", "price", "stock", "expiryDate"],
      properties: {
        medId: { bsonType: "int" },
        medName: { bsonType: "string" },
        supplierId: { bsonType: "int" },
        price: { bsonType: "double" },
        stock: { bsonType: "int", minimum: 0 },
        expiryDate: { bsonType: "date" }
```

```
            }
        }
    }
})

db.createCollection("suppliers", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["supplierId", "name", "city"],
            properties: {
                supplierId: { bsonType: "int" },
                name: { bsonType: "string" },
                city: { bsonType: "string" }
            }
        }
    }
})

db.createCollection("orders", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["orderId", "custId", "medId", "quantity", "orderDate"],
            properties: {
                orderId: { bsonType: "int" },
                custId: { bsonType: "int" },
                medId: { bsonType: "int" },
                quantity: { bsonType: "int", minimum: 1 },
                orderDate: { bsonType: "date" }
            }
        }
    }
})

db.createCollection("pharmacists", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["pharmaId", "name", "joiningDate"],
            properties: {
                pharmaId: { bsonType: "int" },
                name: { bsonType: "string" },
                joiningDate: { bsonType: "date" }
            }
        }
    }
})
```

c. Insert sample data (10 docs each)

```
// Customers
db.customers.insertMany([
  { custId: 1, name: "Arjun", age: 25, phone: "9876543210" },
  { custId: 2, name: "Priya", age: 30, phone: "9876501234" },
  { custId: 3, name: "Ravi", age: 22, phone: "9876512345" },
  { custId: 4, name: "Meena", age: 28, phone: "9876523456" },
  { custId: 5, name: "Amit", age: 35, phone: "9876534567" },
  { custId: 6, name: "Kiran", age: 40, phone: "9876545678" },
  { custId: 7, name: "Sita", age: 29, phone: "9876556789" },
  { custId: 8, name: "Raj", age: 33, phone: "9876567890" },
  { custId: 9, name: "Anita", age: 26, phone: "9876578901" },
  { custId: 10, name: "Vikram", age: 38, phone: "9876589012" }
])

// Suppliers
db.suppliers.insertMany([
  { supplierId: 1, name: "MedLife Pvt Ltd", city: "Bangalore" },
  { supplierId: 2, name: "PharmaCare", city: "Chennai" },
  { supplierId: 3, name: "MediWorld", city: "Delhi" },
  { supplierId: 4, name: "HealthPlus", city: "Hyderabad" },
  { supplierId: 5, name: "LifeMeds", city: "Mumbai" },
  { supplierId: 6, name: "BioPharma", city: "Kolkata" },
  { supplierId: 7, name: "CareWell", city: "Bangalore" },
  { supplierId: 8, name: "Wellness Drugs", city: "Pune" },
  { supplierId: 9, name: "MediCure", city: "Ahmedabad" },
  { supplierId: 10, name: "GoodHealth Pharma", city: "Jaipur" }
])

// Medicines
db.medicines.insertMany([
  { medId: 1, medName: "Paracetamol", supplierId: 1, price: 20.0, stock: 100, expiryDate: ISODate("2025-12-31") },
  { medId: 2, medName: "Amoxicillin", supplierId: 2, price: 50.0, stock: 80, expiryDate: ISODate("2025-06-30") },
  { medId: 3, medName: "Ibuprofen", supplierId: 3, price: 30.0, stock: 60, expiryDate: ISODate("2025-11-15") },
  { medId: 4, medName: "Cetirizine", supplierId: 4, price: 15.0, stock: 120, expiryDate: ISODate("2025-09-10") },
  { medId: 5, medName: "Azithromycin", supplierId: 5, price: 60.0, stock: 50, expiryDate: ISODate("2025-08-25") },
  { medId: 6, medName: "Vitamin C", supplierId: 6, price: 10.0, stock: 200, expiryDate: ISODate("2026-01-20") },
  { medId: 7, medName: "Dolo 650", supplierId: 7, price: 25.0, stock: 150, expiryDate: ISODate("2025-07-15") },
  { medId: 8, medName: "Metformin", supplierId: 8, price: 40.0, stock: 90, expiryDate:
ISODate("2025-10-05") },
  { medId: 9, medName: "Atorvastatin", supplierId: 9, price: 35.0, stock: 70,
expiryDate: ISODate("2025-11-30") },
  { medId: 10, medName: "Omeprazole", supplierId: 10, price: 45.0, stock: 65,
expiryDate: ISODate("2025-12-10") }
])

// Pharmacists
db.pharmacists.insertMany([
  { pharmaId: 1, name: "Dr. Kumar", joiningDate: ISODate("2018-06-10") },
  { pharmaId: 2, name: "Dr. Meera", joiningDate: ISODate("2017-07-15") },
  { pharmaId: 3, name: "Dr. Ramesh", joiningDate: ISODate("2019-03-20") },
  { pharmaId: 4, name: "Dr. Anita", joiningDate: ISODate("2016-08-25") },
  { pharmaId: 5, name: "Dr. Vijay", joiningDate: ISODate("2020-01-12") },
  { pharmaId: 6, name: "Dr. Sunita", joiningDate: ISODate("2021-09-05") },
  { pharmaId: 7, name: "Dr. Karthik", joiningDate: ISODate("2015-11-11") },
  { pharmaId: 8, name: "Dr. Rekha", joiningDate: ISODate("2019-02-14") },
  { pharmaId: 9, name: "Dr. Arvind", joiningDate: ISODate("2020-05-18") },
  { pharmaId: 10, name: "Dr. Nisha", joiningDate: ISODate("2022-04-21") }
])

// Orders
db.orders.insertMany([
  { orderId: 1, custId: 1, medId: 1, quantity: 2, orderDate: ISODate("2025-08-01") },
  { orderId: 2, custId: 1, medId: 2, quantity: 1, orderDate: ISODate("2025-08-02") },
  { orderId: 3, custId: 2, medId: 3, quantity: 3, orderDate: ISODate("2025-08-03") },
  { orderId: 4, custId: 3, medId: 4, quantity: 1, orderDate: ISODate("2025-08-04") },
  { orderId: 5, custId: 4, medId: 5, quantity: 2, orderDate: ISODate("2025-08-05") },
  { orderId: 6, custId: 5, medId: 6, quantity: 4, orderDate: ISODate("2025-08-06") },
  { orderId: 7, custId: 6, medId: 7, quantity: 1, orderDate: ISODate("2025-08-07") },
  { orderId: 8, custId: 7, medId: 8, quantity: 2, orderDate: ISODate("2025-08-08") },
  { orderId: 9, custId: 8, medId: 9, quantity: 3, orderDate: ISODate("2025-08-09") },
  { orderId: 10, custId: 9, medId: 10, quantity: 1, orderDate: ISODate("2025-08-10") }
])
```

d. Aggregation queries

i. List all orders with customer name and medicine name

```
db.orders.aggregate([
  { $lookup: {
    from: "customers",
    localField: "custId",
    foreignField: "custId",
    as: "customerInfo"
  }},
  { $unwind: "$customerInfo" },
  { $lookup: {
    from: "medicines",
    localField: "medId",
    foreignField: "medId",
    as: "medicineInfo"
  }},
  { $unwind: "$medicineInfo" },
  { $project: { _id: 0, orderId: 1, customer: "$customerInfo.name", medicine: "$medicineInfo.medName", quantity: 1, pricePerUnit: "$medicineInfo.price" } }
])
```

ii. Total sales per medicine

```
db.orders.aggregate([
  { $lookup: {
    from: "medicines",
    localField: "medId",
    foreignField: "medId",
    as: "medInfo"
  }},
  { $unwind: "$medInfo" },
  { $group: {
    _id: "$medInfo.medName",
    totalSold: { $sum: "$quantity" },
    totalRevenue: { $sum: { $multiply: ["$quantity", "$medInfo.price"] } }
  }},
  { $sort: { totalRevenue: -1 } }
])
```

iii. Supplier-wise revenue

```
db.orders.aggregate([
  { $lookup: {
    from: "medicines",
    localField: "medId",
    foreignField: "medId",
    as: "medInfo"
  }},
  { $unwind: "$medInfo" },
  { $lookup: {
    from: "suppliers",
    localField: "medInfo.supplierId",
    foreignField: "supplierId",
    as: "supplierInfo"
  }},
  { $unwind: "$supplierInfo" },
  { $group: {
    _id: "$supplierInfo.name",
    totalRevenue: { $sum: { $multiply: ["$quantity", "$medInfo.price"] } }
  }},
  { $sort: { totalRevenue: -1 } }
])
```

iv. **Medicines below stock threshold (e.g., < 70 units)**

```
db.medicines.aggregate([
  { $match: { stock: { $lt: 70 } } },
  { $lookup: {
      from: "suppliers",
      localField: "supplierId",
      foreignField: "supplierId",
      as: "supplierInfo"
  }},
  { $unwind: "$supplierInfo" },
  { $project: { _id: 0, medName: 1, stock: 1, supplier: "$supplierInfo.name" } }
])
```

v. **Customer order summary**

```
db.customers.aggregate([
  { $lookup: {
      from: "orders",
      localField: "custId",
      foreignField: "custId",
      as: "ordersList"
  }},
  { $project: {
      _id: 0,
      name: 1,
      totalOrders: { $size: "$ordersList" }
  }},
  { $sort: { totalOrders: -1 } }
])
```

Structured Enquiry

Lab Program4:

2. Create patient database with 5 collections: **patients**; **doctors; medicines; prescriptions**; **sales** Schema validation will ensure fields like **age** are within a valid range, **dates** follow ISODate format, and **IDs** are unique. Then create different indexes and run queries to show their use.

a. Create Database and Collections with Validation

b. Insert Sample Data (10 Documents per Collection)

c. Creating Indexes

  i. Single Field Index

  ii. Compound Index

  iii. Text Index

  iv. Partial Index

  v. Unique Index

d. Example Queries Using Indexes

e. Index Verification

f. Performance Comparison Example

   i. Without Index

   ii. With Index

a. Create Database and Collections with Validation
use hospitalDB

```
//Patients Collection (Validation: age 0-120, dateOfBirth must be date)
db.createCollection("patients", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["patientId", "name", "age", "dateOfBirth"],
      properties: {
        patientId: { bsonType: "int" },
        name: { bsonType: "string" },
        age: { bsonType: "int", minimum: 0, maximum: 120 },
        dateOfBirth: { bsonType: "date" }
      }
    }
  }
})

//Doctors Collection
db.createCollection("doctors", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["doctorId", "name", "specialization"],
      properties: {
        doctorId: { bsonType: "int" },
        name: { bsonType: "string" },
        specialization: { bsonType: "string" }
      }
    }
  }
})

//Medicines Collection (expiryDate as date)
db.createCollection("medicines", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["medicineId", "name", "category", "price", "expiryDate"],
      properties: {
        medicineId: { bsonType: "int" },
        name: { bsonType: "string" },
                category: { bsonType: "string" },
                price: { bsonType: "double" },
                expiryDate: { bsonType: "date" }
            }
        }
    }
})

//Prescriptions Collection
db.createCollection("prescriptions", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["prescriptionId", "patientId", "doctorId", "medicineId", "dateIssued"],
            properties: {
                prescriptionId: { bsonType: "int" },
                patientId: { bsonType: "int" },
                doctorId: { bsonType: "int" },
                medicineId: { bsonType: "int" },
                dateIssued: { bsonType: "date" }
            }
        }
    }
})

//Sales Collection
db.createCollection("sales", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["saleId", "medicineId", "quantity", "saleDate"],
            properties: {
                saleId: { bsonType: "int" },
                medicineId: { bsonType: "int" },
                quantity: { bsonType: "int" },
                saleDate: { bsonType: "date" }
            }
        }
    }
})
```

b. Insert Sample Data (10 Documents per Collection) use hospitalDB

```
db.patients.insertMany([
  { patientId: 1, name: "Alice Johnson", age: 29, dateOfBirth: new Date("1996-03-15") },
  { patientId: 2, name: "Bob Smith", age: 45, dateOfBirth: new Date("1980-07-22") },
  { patientId: 3, name: "Charlie Lee", age: 60, dateOfBirth: new Date("1965-01-10") },
  { patientId: 4, name: "David Kim", age: 34, dateOfBirth: new Date("1991-11-03") },
  { patientId: 5, name: "Eva Brown", age: 51, dateOfBirth: new Date("1974-05-09") },
  { patientId: 6, name: "Frank Green", age: 72, dateOfBirth: new Date("1953-02-14") },
  { patientId: 7, name: "Grace White", age: 40, dateOfBirth: new Date("1985-06-17") },
  { patientId: 8, name: "Henry Miller", age: 25, dateOfBirth: new Date("2000-08-20") },
  { patientId: 9, name: "Ivy Davis", age: 37, dateOfBirth: new Date("1988-09-29") },
  { patientId: 10, name: "Jack Wilson", age: 55, dateOfBirth: new Date("1970-04-05") }
])

db.doctors.insertMany([
  { doctorId: 1, name: "Dr. Emily Carter", specialization: "Cardiology" },
  { doctorId: 2, name: "Dr. Michael Brown", specialization: "Neurology" },
  { doctorId: 3, name: "Dr. Sophia Lee", specialization: "Pediatrics" },
  { doctorId: 4, name: "Dr. David Chen", specialization: "Orthopedics" },
  { doctorId: 5, name: "Dr. Laura White", specialization: "Dermatology" },
  { doctorId: 6, name: "Dr. James Wilson", specialization: "Psychiatry" },
  { doctorId: 7, name: "Dr. Olivia Smith", specialization: "Gynecology" },
  { doctorId: 8, name: "Dr. Robert Clark", specialization: "Oncology" },
  { doctorId: 9, name: "Dr. Linda Johnson", specialization: "Ophthalmology" },
  { doctorId: 10, name: "Dr. William Scott", specialization: "ENT" }
])

db.medicines.insertMany([
  { medicineId: 1, name: "Paracetamol", category: "Pain Relief", price: 2.5, expiryDate: new Date("2026-05-01") },
  { medicineId: 2, name: "Amoxicillin", category: "Antibiotic", price: 5.0, expiryDate: new Date("2025-08-15") },
  { medicineId: 3, name: "Cetirizine", category: "Antihistamine", price: 1.5, expiryDate: new Date("2027-01-10") },
  { medicineId: 4, name: "Ibuprofen", category: "Pain Relief", price: 3.0, expiryDate: new Date("2026-10-20") },
  { medicineId: 5, name: "Metformin", category: "Diabetes", price: 4.0, expiryDate: new Date("2025-12-05") },
  { medicineId: 6, name: "Aspirin", category: "Blood Thinner", price: 2.0, expiryDate: new Date("2027-03-12") },
  { medicineId: 7, name: "Atorvastatin", category: "Cholesterol", price: 6.0, expiryDate: new Date("2025-07-30") },
  { medicineId: 8, name: "Omeprazole", category: "Acidity", price: 3.5, expiryDate: new Date("2026-09-25") },
  { medicineId: 9, name: "Salbutamol", category: "Asthma", price: 7.0, expiryDate: new Date("2025-11-11") },
  { medicineId: 10, name: "Vitamin D", category: "Supplement", price: 4.5, expiryDate: new Date("2027-02-19") }
])

db.prescriptions.insertMany([
  { prescriptionId: 1, patientId: 1, doctorId: 1, medicineId: 1, dateIssued: new Date("2025-08-01") },
  { prescriptionId: 2, patientId: 2, doctorId: 2, medicineId: 2, dateIssued: new Date("2025-08-02") },
  { prescriptionId: 3, patientId: 3, doctorId: 3, medicineId: 3, dateIssued: new Date("2025-08-03") },
  { prescriptionId: 4, patientId: 4, doctorId: 4, medicineId: 4, dateIssued: new Date("2025-08-04") },
  { prescriptionId: 5, patientId: 5, doctorId: 5, medicineId: 5, dateIssued: new Date("2025-08-05") },
  { prescriptionId: 6, patientId: 6, doctorId: 6, medicineId: 6, dateIssued: new Date("2025-08-06") },
  { prescriptionId: 7, patientId: 7, doctorId: 7, medicineId: 7, dateIssued: new Date("2025-08-07") },
  { prescriptionId: 8, patientId: 8, doctorId: 8, medicineId: 8, dateIssued: new Date("2025-08-08") },
  { prescriptionId: 9, patientId: 9, doctorId: 9, medicineId: 9, dateIssued: new Date("2025-08-09") },
  { prescriptionId: 10, patientId: 10, doctorId: 10, medicineId: 10, dateIssued: new Date("2025-08-10") }
])

db.sales.insertMany([
  { saleId: 1, medicineId: 1, quantity: 5, saleDate: new Date("2025-08-01") },
  { saleId: 2, medicineId: 2, quantity: 3, saleDate: new Date("2025-08-02") },
  { saleId: 3, medicineId: 3, quantity: 2, saleDate: new Date("2025-08-03") },
  { saleId: 4, medicineId: 4, quantity: 4, saleDate: new Date("2025-08-04") },
  { saleId: 5, medicineId: 5, quantity: 1, saleDate: new Date("2025-08-05") },
  { saleId: 6, medicineId: 6, quantity: 6, saleDate: new Date("2025-08-06") },
  { saleId: 7, medicineId: 7, quantity: 3, saleDate: new Date("2025-08-07") },
  { saleId: 8, medicineId: 8, quantity: 2, saleDate: new Date("2025-08-08") },
  { saleId: 9, medicineId: 9, quantity: 5, saleDate: new Date("2025-08-09") },
  { saleId: 10, medicineId: 10, quantity: 4, saleDate: new Date("2025-08-10") }
])
```

c. Creating Indexes

i. Single Field Index

```
// Index on patientId for quick retrieval
db.patients.createIndex({ patientId: 1 })
```

ii. Compound Index

```
// Search prescriptions by patientId and dateIssued
db.prescriptions.createIndex({ patientId: 1, dateIssued: -1 })
```

iii. **Text Index**

```
// Search medicines by name or category
db.medicines.createIndex({ name: "text", category: "text" })
// Example query: find medicines with keyword "pain"
db.medicines.find({ $text: { $search: "pain" } })
```

iv. **Partial Index**

```
// Index only for medicines that are not expired
db.medicines.createIndex(
   { expiryDate: 1 }, { partialFilterExpression: { expiryDate: { $gte: new Date() } } })
```

v. **Unique Index**

```
// Ensure each doctor has a unique doctorId
db.doctors.createIndex({ doctorId: 1 }, { unique: true })
```

j. **Example Queries Using Indexes**

```
// Single field index usage
db.patients.find({ patientId: 5 })
// Compound index usage
db.prescriptions.find({ patientId: 3 }).sort({ dateIssued: -1 })
// Text search usage
db.medicines.find({ $text: { $search: "antibiotic" } })
// Partial index usage
db.medicines.find({ expiryDate: { $gte: new Date() } })
// Unique index check (will throw error if duplicate doctorId inserted)
db.doctors.insertOne({ doctorId: 1, name: "Duplicate", specialization: "Test" })
```

k. **Index Verification**

```
//View all indexes in medicines collection
db.medicines.getIndexes()
```

l. **Performance Comparison Example**

i. **Without Index**

```
// Drop any existing index for a clean test
db.patients.dropIndexes()
// Query to find patient with ID 7 (no index yet)
db.patients.find({ patientId: 7 }).explain("executionStats")
```

ii. **With Index**

```
// Create single field index on patientId
db.patients.createIndex({ patientId: 1 })
// Run the same query again
db.patients.find({ patientId: 7 }).explain("executionStats")
```