**Big Data Analytics**

**BAD601**

**Experiment 5:**

Implement Functions: Count – Sort – Limit – Skip – Aggregate using MongoDB

Open Command Prompt as Administrator and type:

mongod

mongosh

1. Create a Database

In MongoDB, a database is created automatically when you insert a document into a collection.

use university

This switches to (or creates if not existing) a database named university.

2. Create a Collection and Insert Documents

Create a students collection and insert multiple documents.

db.students.insertMany([

{ name: "Alice", age: 22, department: "CS", marks: 85 },

{ name: "Bob", age: 24, department: "IT", marks: 78 },

{ name: "Charlie", age: 21, department: "CS", marks: 90 },

{ name: "David", age: 23, department: "IT", marks: 88 },

{ name: "Eve", age: 20, department: "ECE", marks: 75 },

{ name: "Frank", age: 25, department: "CS", marks: 92 },

{ name: "Grace", age: 22, department: "ECE", marks: 80 }

])

3. Count Documents

The countDocuments() method counts documents that match a query.

db.students.countDocuments({ age: { $gte: 22 } })

🔹 Counts the number of students with age greater than or equal to 22.

4. Sort Documents

Sorting documents in ascending (1) or descending (-1) order.

db.students.find().sort({ marks: -1 })

🔹 Sorts students in descending order of marks.

5. Limit the Number of Results

The limit() method restricts the number of documents retrieved.

db.students.find().limit(3)

🔹 Retrieves only the first 3 student documents.

6. Skip Documents

The skip() method allows skipping a specified number of documents.

db.students.find().skip(2)

🔹 Skips the first 2 student records and returns the rest.

7. Aggregate Function

Aggregation allows data transformation and complex calculations.

db.students.aggregate([

{ $match: { marks: { $gte: 80 } } }, // Filter students with marks >= 80

{ $group: { \_id: "$department", avgMarks: { $avg: "$marks" } } }, // Group by department, calculate avg marks

{ $sort: { avgMarks: -1 } }, // Sort by average marks in descending order

{ $limit: 2 } // Return top 2 departments with highest average marks

])

🔹 Finds the average marks for each department, sorts them, and returns the top 2.