ASSIGNMENT 2 - MONGODB (21.07.25)

Installation Process :

Step 1: Download MongoDB Installer

1. Go to the official MongoDB website:
   https://www.mongodb.com/try/download/community
2. Under "Version", choose the latest stable release.
3. Make sure:
   - Platform is set to Windows
   - Package is set to MSI
4. Click the Download button.

Step 2: Run the Installer

1. Once the .msi file is downloaded, double-click it.
2. Click Next on the setup screen.
3. Choose "Complete" installation.

Step 3: Install MongoDB as a Windows Service

1. Keep the default settings for:
   - Run MongoDB as a Service
   - Service Name: `MongoDB`

2. Make sure "Install MongoDB Compass" is checked (optional but helpful GUI).

3. Click Next, then Install.

4. Wait for installation to complete and click Finish.

Step 4: Set MongoDB Path in Environment Variables (optional but recommended)

1. Go to Start → search for Environment Variables → click "Edit the system environment variables".

2. In the System Properties window:

    - Click Environment Variables

3. Under System variables, find and click on Path, then click Edit

Click New, then paste the MongoDB bin path :

C:\Program Files\MongoDB\Server\<version>\bin

4. Replace <version> with your installed version

5. Click OK to save.

Step 5: Create MongoDB Data Folder

MongoDB stores data in a folder. You need to create it:

1. Open **C:\ drive.**

2. Create a new folder:
   C:\data\db
3. If needed, create the folders manually one by one (data, then db inside it).

Step 6: Run MongoDB

1. Open command prompt

2. Start the MongoDB server:
   mongod

3. Open a new Command Prompt window and type:
   mongo

This opens the MongoDB shell (you can now run MongoDB commands).

CRUD Operations :

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\masih>mongosh
Current Mongosh Log ID: 687e03a8bd5b301f2feec4a8
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.6
Using MongoDB:          8.0.11
Using Mongosh:          2.5.6

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-07-21T12:53:07.682+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------
```

Database Operation:

```
test> show dbs
admin   40.00 KiB
amitdb   8.00 KiB
config  72.00 KiB
local   40.00 KiB
```

Collection Operation:

```
test> use jisl
switched to db jisl
jisl> db
jisl
jisl> show dbs
admin   40.00 KiB
amitdb   8.00 KiB
config  96.00 KiB
local   40.00 KiB
```

A. Create Operations :

1. insertOne() – Insert a Single Document

```
jisl> db.inventory.insertOne({ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } })
{
  acknowledged: true,
  insertedId: ObjectId('687e0926bd5b301f2feec4a9')
}
```

2. insertMany() – Insert Multiple Documents at Once

```
jisl> db.inventory.insertMany([{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },{ item: "mat", qty: 85, tags: ["gray"
], size: { h: 27.9, w: 35.5, uom: "cm" } }, { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('687e0935bd5b301f2feec4aa'),
    '1': ObjectId('687e0935bd5b301f2feec4ab'),
    '2': ObjectId('687e0935bd5b301f2feec4ac')
  }
}
```

```
jisl> show collections
inventory
user
```

B. Read Operations

MongoDB find() Statements

1. Find all documents in inventory

```
jisl> db.inventory.find()
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

2. Project specific fields (item, qty)

```
jisl> db.inventory.find({}, { item: 1, qty: 1 })
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25
  },
  { _id: ObjectId('687e0935bd5b301f2feec4ab'), item: 'mat', qty: 85 },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25
  }
]
```

3. Exclude _id and include only item and qty

```
jisl> db.inventory.find({}, { item: 1, qty: 1, _id: 0 })
[
  { item: 'canvas', qty: 100 },
  { item: 'journal', qty: 25 },
  { item: 'mat', qty: 85 },
  { item: 'mousepad', qty: 25 }
]
jisl> db.inventory.find({ tags: "gray" })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  }
]
jisl> db.inventory.find({ tags: { $ne: "gray" } })
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

4. Find documents with a specific tag ("gray")

```
jisl> db.inventory.find({ tags: "gray" })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  }
]
```

5. Find documents where tag is not "gray"

```
jisl> db.inventory.find({ tags: { $ne: "gray" } })
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

6. Find documents where size.h is 28 and size.uom is 'cm'

```
jisl> db.inventory.find({ "size.h": 28, "size.uom": "cm" })
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  }
]
```

7. Find documents where qty is 25 OR item is "mat"

```
jisl> db.inventory.find({
...   $or: [
...     { qty: 25 },
...     { item: "mat" }
...   ]
... })
...
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

1. Find items with qty greater than 25

```
jisl> db.inventory.find({ qty: { $gt: 25 } })
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  }
]
```

2. Find items with qty between 25 and 50

```
jisl> db.inventory.find({ qty: { $gte: 25, $lte: 50 } })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

3. Find items where item field contains "jo" (regex match)

```
jisl> db.inventory.find({ item: /jo/ })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  }
]
```

4. Explicitly with $regex Or string-based regex:

```
jisl> db.inventory.find({ item: { $regex: /jo/ } })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  }
]
jisl> db.inventory.find({ item: { $regex: "^jo" } })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  }
]
```

5. Sort results by qty ascending

```
jisl> db.inventory.find().sort({ qty: 1 })
[
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  }
]
```

6. Sort results by qty descending

```
jisl> db.inventory.find().sort({ qty: -1 })
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ab'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4aa'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('687e0935bd5b301f2feec4ac'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

7. Count total documents

```
jisl> db.inventory.countDocuments()
4
jisl> db.inventory.find().count()
4
```

8. Count items where tags field exists

```
jisl> db.inventory.countDocuments({ tags: { $exists: true } })
4
```

9. Count items with qty exactly 30

```
jisl> db.inventory.countDocuments({ qty: 30 })
0
```

10. Count items with qty greater than 30

```
jisl> db.inventory.countDocuments({ qty: { $gt: 30 } })
2
```

11. Get distinct values from the tags field

```
jisl> db.inventory.distinct("tags")
[ 'blank', 'blue', 'cotton', 'gel', 'gray', 'red' ]
```

12. Find one item (any random document)

```
jisl> db.inventory.findOne()
{
  _id: ObjectId('687e0926bd5b301f2feec4a9'),
  item: 'canvas',
  qty: 100,
  tags: [ 'cotton' ],
  size: { h: 28, w: 35.5, uom: 'cm' }
}
```

13. Limit to 1 result

```
jisl> db.inventory.find().limit(1)
[
  {
    _id: ObjectId('687e0926bd5b301f2feec4a9'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' },
    lastModified: ISODate('2025-07-21T10:55:54.307Z'),
    status: 'P'
  }
]
```

14. Explain query plan for finding items with qty: 25

```
jisl> db.inventory.find({ qty: 25 }).explain()
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'jisl.inventory',
    parsedQuery: { qty: { '$eq': 25 } },
    indexFilterSet: false,
    queryHash: 'FD30B09C',
    planCacheShapeHash: 'FD30B09C',
    planCacheKey: 'A5384E34',
    optimizationTimeMillis: 0,
```

```
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { qty: { '$eq': 25 } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  queryShapeHash: '47D7080DE0C83A417C81A922225B16C421E731E485B63FFE8594BFCEC
6D0270C',
  command: { find: 'inventory', filter: { qty: 25 }, '$db': 'jisl' },
  serverInfo: {
    host: 'LAPTOP-N9BPVIB6',
    port: 27017,
    version: '8.0.11',
    gitVersion: 'bed99f699da6cb2b74262aa6d473446c41476643'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted',
    internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
  },
  ok: 1
}
```

B. Update Operations

1. updateOne() – Update the first matching document

```
jisl> db.inventory.updateOne(
...    { item: "canvas" },
...    {
...      $set: { "size.uom": "cm", status: "P" },
...      $currentDate: { lastModified: true }
...    }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2. updateMany() – Update all matching documents

```
jisl> db.inventory.updateMany(
...    { qty: { $lt: 50 } },
...    {
...      $set: { status: "Low Stock" },
...      $currentDate: { lastModified: true }
...    }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

3. replaceOne() – Completely replace the document

```
jisl> db.inventory.replaceOne(
...    { item: "mousepad" },
...    {
...      item: "mousepad",
...      qty: 30,
...      tags: ["black", "rubber"],
...      size: { h: 20, w: 25, uom: "cm" }
...    }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Delete Operations

1. deleteOne() – Delete the first matching document

```
jisl> db.inventory.deleteOne({ status: "D" })
{ acknowledged: true, deletedCount: 0 }
```

2. deleteMany() – Delete all matching documents

```
jisl> db.inventory.deleteMany({ status: "D" })
{ acknowledged: true, deletedCount: 0 }
```