EXP NO :8 DATE:	**Capped Collections**

AIM

To write Conditional and Range queries for managing the To-Do List project using MongoDB, focusing on filtering and organizing tasks.

QUERIES

1. Retrieve all completed tasks:
    - Query: db.tasks.find({"status": "completed"})
    - Purpose: Retrieve all tasks marked as "completed."
    - Example Output:
    - { _id: 101, task_name: 'Buy Groceries', status: 'completed' }
    - { _id: 102, task_name: 'Pay Bills', status: 'completed' }

2. Retrieve tasks with priority 'High' or 'Medium':
    - Query: db.tasks.find({"priority": {"$in": ["High", "Medium"]}})
    - Purpose: Filter tasks with specific priority levels.
    - Example Output:
    - { _id: 103, task_name: 'Submit Assignment', priority: 'High' }
    - { _id: 104, task_name: 'Clean Room', priority: 'Medium' }

3. Retrieve tasks with a deadline within the next 7 days and marked as 'In Progress':
    - Query: db.tasks.find({"$and": [{"status": "In Progress"}, {"deadline": {"$lt": ISODate("2025-04-07")}}]})
    - Purpose: Identify urgent, in-progress tasks.
    - Example Output:
    - { _id: 105, task_name: 'Prepare Presentation', deadline: '2025-04-05', status: 'In Progress' }

4. Retrieve tasks labeled as 'Low Priority' or with a deadline more than 30 days away:
    - Query: db.tasks.find({"$or": [{"priority": "Low"}, {"deadline": {"$gt": ISODate("2025-05-01")}}]})
    - Purpose: Find non-urgent tasks or those with relaxed deadlines.
    - Example Output:
    - { _id: 106, task_name: 'Organize Photos', priority: 'Low', deadline: '2025-06-01' }

5. Retrieve tasks with tags including 'Urgent' and either a priority higher than 'Low' or due within a day:
    - Query: db.tasks.find({"$and": [{"tags": "Urgent"}, {"$or": [{"priority": {"$ne": "Low"}}, {"deadline": {"$lte": ISODate("2025-04-02")}}]}]})
    - Purpose: Pinpoint urgent tasks requiring immediate attention.
    - Example Output:
    - { _id: 107, task_name: 'Repair Laptop', tags: ['Urgent'], deadline: '2025-04-01' }

6. Exclude tasks in the 'Completed' or 'Cancelled' statuses:
    - Query: db.tasks.find({"$and": [{"status": {"$ne": "Completed"}}, {"status": {"$ne": "Cancelled"}}]})
    - Purpose: Focus on active tasks.
    - Example Output:
    - { _id: 108, task_name: 'Plan Trip', status: 'In Progress' }

7. Retrieve tasks with less than 3 subtasks:
    - Query: db.tasks.find({"subtasks_count": {"$not": {"$gt": 3}}})
    - Purpose: Filter small tasks.
    - Example Output:
    - { _id: 109, task_name: 'Update CV', subtasks_count: 2 }

8. Find tasks marked as 'High Priority' and with an estimated completion time of 2 hours:
    - Query: db.tasks.find({"$and": [{"priority": "High"}, {"estimated_time": 2}]})

- Purpose: Focus on short, high-priority tasks.
- Example Output:
- { _id: 110, task_name: 'Write Blog Post', estimated_time: 2, priority: 'High' }

9. Retrieve tasks with due dates within the next month and marked as either 'Important' or tagged 'Personal':
    - Query: db.tasks.find({"deadline": {"$lte": ISODate("2025-05-01")}, "$or": [{"tags": "Important"}, {"tags": "Personal"}]})
    - Purpose: Identify tasks due soon and relevant to personal priorities.
    - Example Output:
    - { _id: 111, task_name: 'File Taxes', deadline: '2025-04-15', tags: ['Important'] }

10. Exclude tasks created by a specific user (e.g., 'Rahul'):
    - Query: db.tasks.find({"created_by": {"$ne": "Rahul"}})
    - Purpose: Filter tasks created by others.
    - Example Output:
    - { _id: 112, task_name: 'Build Portfolio', created_by: 'Aditi' }

11. Retrieve tasks titled 'Morning Jog' or 'Yoga Session':
    - Query: db.tasks.find({"task_name": {"$in": ["Morning Jog", "Yoga Session"]}})
    - Purpose: Focus on specific health-related tasks.
    - Example Output:
    - { _id: 113, task_name: 'Morning Jog' }
    - { _id: 114, task_name: 'Yoga Session' }

12. Retrieve tasks with time estimates of 30 or 60 minutes:
    - Query: db.tasks.find({"estimated_time": {"$in": [30, 60]}})
    - Purpose: Find short, timed tasks.
    - Example Output:
    - { _id: 115, task_name: 'Meditate', estimated_time: 30 }

- { _id: 116, task_name: 'Quick Workout', estimated_time: 60 }

13. Retrieve tasks created for the 'Chennai Office':
    - Query: db.tasks.find({"branch.location": "Chennai"})
    - Purpose: Locate tasks related to a specific office.
    - Example Output:
    - { _id: 117, task_name: 'Prepare Chennai Presentation', branch: { location: 'Chennai' } }

CLASS PERFORMANCE	
VIVA	
RECORD	
TOTAL	

RESULT

Hence,Conditional and range queries for the To-Do List project using MongoDB are now designed, implemented, and verified.

EXP NO:9 DATE:	GEOSPATIAL INDEXES IN MONGODB

AIM:

To write queries for geospatial indexes for To-Do List Management using MongoDB.

GEOSPATIAL QUERIES:

MongoDB supports query operations on geospatial data. Geospatial indexes are used to calculate geometry over an Earth-like sphere. A field named type specifies the GeoJSON object type.

Syntax:

<field>: {
 type: <GeoJSON type>,
 coordinates: <coordinates>
}

2D SPHERE:

2D Sphere indexes support queries that calculate geometries on a sphere. To create an index for 2dsphere:
db.tasks.createIndex({<location field>: "2dsphere"})

QUERYING ON GEOSPATIAL DATA:

MongoDB provides the following geospatial query operators:

1. **$geoIntersects:**
   - Selects geometries that intersect with a GeoJSON geometry.
   - Supported by 2dSphere index.
2. **$within:**
   - Selects geometries within a bounding GeoJSON geometry.
3. **$near:**
   - Returns geospatial objects in proximity to a point.
   - Requires a geospatial index.
   - 2dSphere indexes support $near.

COLLECTIONS FOR GEOSPATIAL INDEX:

1. **Creating Point Type:**
   Query: db.tasks.insertMany([
   {loc: {type: "Point", coordinates: [-73.88, 40.78]}, task_name: "Meeting with

Client"},
{loc: {type: "Point", coordinates: [-122.42, 37.77]}, task_name: "Grocery Shopping"}
])

Output:

{ "acknowledged": true, "insertedIds": [
ObjectId("627e2cf421ae3b4b52468d0f"),
ObjectId("627e2cf421ae3b4b52468d10")] }

2. **Creating Line Type:**

Query: db.tasks.insert({
loc2: {type: "LineString", coordinates: [[0,1], [0,2], [1,2]]}, task_name: "Jogging Route"
})

Output:

WriteResult({"nInserted": 1 })

3. **Creating Polygon Type:**

Query: db.tasks.insert({
loc3: {type: "Polygon", coordinates: [[-73.99, 40.72]]}, task_name: "Work Zone"
})

Output:

WriteResult({ "nInserted": 1 })

4. **Creating 2D Sphere Type:**

Query: db.tasks.insert({
loc4: {type: "2dsphere", coordinates: [[0,0], [1,2], [2,3]]}, task_name: "Daily Commute Path"
})

Output:

WriteResult({ "nInserted": 1 })

TO CREATE A 2D SPHERE INDEX:

Query: db.tasks.createIndex({loc: "2dsphere"})

Output:

{ "numIndexesBefore": 1, "numIndexesAfter": 2, "createdCollectionAutomatically": false, "ok": 1 }

QUERYING ON GEOSPATIAL INDEXING:

1. **Using $geoIntersects:**

Query: db.tasks.find({ loc: {
$geoIntersects: {
$geometry: {type: "Polygon", coordinates: [[-73.99, 40.72]]}
}
}})

Output:


```
{
  "id": "6295aedf13da7654db012",
  "geometry": {
    "coordinates": [[-73.93,40.81], [-73.93,40.81]],
    "type": "Polygon"
  },
  "task_name": "Office Area"
}
```


2. **Using $geoWithin:**

Query: db.tasks.find({ loc2: {
$geoWithin: {
$geometry: {type: "Polygon", coordinates: [[-73.97, 40.82]]}
}
}})

Output:


```
{
  "_id": ObjectId("6242cd6700ca0b233e84a2a2"),
  "location": {
    "coordinates": [-73.97, 40.82],
    "type": "Point"
  },
  "task_name": "Morning Walk"
}
```


3. **Using $near:**

Query: db.tasks.find({ location: {
$near: {
$geometry: {type: "Point", coordinates: [-73.79, 40.75]},
$maxDistance: 150000,
$minDistance: 500
}
}})

Output:


```
{
  "id": ObjectId("6242cd6700ca0b233e84a2a3"),

 "location": {
     "coordinates": [-73.85, 40.72],
     "type": "Point"
  },
  "task_name": "Nearest Grocery Store"
}
```


CLASS PERFORMANCE	
VIVA	
RECORD	
TOTAL	

RESULT:

Hence, queries for geospatial indexes for To-Do List Management System are implemented and the output is verified.

EXP NO : 11 DATE:	MAP REDUCE AND REPLICATION

AIM:

To implement MapReduce & Replication for Task Management System.

MAP:

It is a JavaScript function that maps a value with a key and emits a key-value pair.

REDUCE:

It is a JavaScript function that reduces or groups all the documents having the same key.

MAPREDUCE:

The mapReduce() function in MongoDB first queries the collection, then maps each document to emit key-value pairs which are then reduced based on the keys that have multiple values.

Syntax:

```
db.collection_name.mapReduce(
  function() { emit(key, value); },
  function(key, values) { return reduceFunction; },
  {
    query: document,
    sort: document,
    limit: document,
    verbose: Boolean,
    out: newCollection
  }
);
```

REPLICATION:

A replica set in MongoDB is a group of mongod processes that maintain the same dataset. Replica sets provide redundancy and high availability, which are essential for all production deployments.

Replication ensures data availability by maintaining multiple copies of data across different servers. This offers fault tolerance against a single server failure.

MAP REDUCE IMPLEMENTATION

Objective:

Calculate total ticket revenue per train route.

1. Create Map and Reduce Functions

```
var map = function() {
  emit(this.route, this.ticketFare);
};


var reduce = function(key, values) {
  return Array.sum(values);
};


db.tickets.mapReduce(
  map,
  reduce,
  { out: "route_revenue_summary" }
);
```

This calculates total revenue per train route (e.g., *Delhi-Mumbai*, *Chennai-Bangalore*, etc.).

DATABASE COLLECTION METRICS:

- collections: 15
- capped: 2
- timeseries: 0
- views: 1

- internalCollections: 2
- internalViews: 0

CONNECTIONS:

{
  "current": 30,
  "available": 999970,
  "totalCreated": 30,
  "active": 4,
  "threaded": 30,
  "exhaustIsMaster": 0,
  "exhaustHello": 3,
  "awaitingTopologyChanges": 1
}

ELECTION METRICS:

{
  "stepUpCmd": { "called": 0, "successful": 0 },
  "priorityTakeover": { "called": 0, "successful": 0 },
  "catchUpTakeover": { "called": 0, "successful": 0 },
  "electionTimeout": { "called": 0, "successful": 0 },
  "freezeTimeout": { "called": 0, "successful": 0 }
}

EXTRA INFO:

{
  "note": "fields vary by platform",
  "page_faults": 40234,
  "usagePageFileMB": 243,
  "totalPageFileMB": 8124,
  "availPageFileMB": 1534,

"ramMB": 8192
}

FLOW CONTROL:

{
"enabled": true,
"targetRateLimit": 1000000000,
"timeAcquiringMicros": 582,
"locksPerKiloOp": 0,
"sustainerRate": 0,
"isLagged": false,
"isLaggedCount": 0,
"isLaggedTimeMicros": 0
}

TRANSACTIONS:

{
"retriedCommandsCount": 0,
"retriedStatementsCount": 0,
"transactionsCollectionWriteCount": 0,
"currentActive": 0,
"currentInactive": 0,
"currentOpen": 0,
"totalAborted": 0,
"totalCommitted": 0,
"totalStarted": 0
}

TRANSPORT SECURITY:

{
"1.0": 0,

```
  "1.1": 0,
  "1.2": 0,
  "1.3": 0,
  "unknown": 0
}
```


TWO PHASE COMMIT COORDINATOR:


```
{
  "totalCreated": 0,
  "totalStartedTwoPhaseCommit": 0,
  "totalAbortedTwoPhaseCommit": 0,
  "totalCommittedTwoPhaseCommit": 0,
  "currentInSteps": {
    "writingParticipantList": 0,
    "waitingForVotes": 0,
    "writingDecision": 0,
    "waitingForDecisionAcks": 0,
    "deleting
```


WIREDTIGER ENGINE STATS:


```
{
  "uri": "statistics:",
  "block-manager":
  {
    "block cache cached blocks updated": 0,
    "block cache cached bytes updated": 0,
    "block cache evicted blocks": 0,
    "block cache file size causing bypass": 0,
    "block cache lookups": 0,
    "block cache number of blocks not evicted due to
```

overhead": 0, "block cache number of bypasses due to overhead on put": 0, "block cache number of hits including existence checks": 0, "block cache number of misses including existence checks": 0
}
}

Class Performance	
Viva	
Record	
Total	

RESULT

Hence, the Map reduce and replication for the Task Management System using MongoDB are implemented and verified.

EX.NO:12 DATE:	GRIDFS IN MONGODB

AIM

To write queries for GridFS for a Task Management System using MongoDB.

GRIDFS IN MONGODB

GridFS (Grid File System) is a specification in MongoDB used for storing and retrieving large files like passenger ID proofs, ticket PDFs, route maps, and other large documents in chunks, especially when they exceed the 16MB BSON document size limit.

Types of GridFS Collections

- fs.files – Stores file metadata like filename, upload date, and chunk size.
- fs.chunks – Store
- s actual file data in binary chunks.

UPLOAD A FILE USING MONGODB SHELL

Query:

```
var bucket = new GridFSBucket(db.getSiblingDB("Task_db"));

var file = fs.openReadStream("/path/to/ticket_pdf.pdf");

bucket.uploadFromStream("ticket_pdf.pdf", file);
```

Output:

File uploaded successfully to GridFS

VIEWING UPLOADED FILES

Query:

```
db.fs.files.find().pretty()
```

Output:

```
{
  "_id": ObjectId("65fd23a4c5a93b12ef456xyz"),
  "filename": "ticket_pdf.pdf",
  "length": 1048576,
  "chunkSize": 261120,
  "uploadDate": ISODate("2025-04-01T12:00:00Z"),
  "metadata": {}
}
```

RETRIEVING THE FILE

Query:

```
var bucket = new GridFSBucket(db.getSiblingDB("Task_db"));
var file = fs.createWriteStream("./downloaded_ticket.pdf");
bucket.openDownloadStreamByName("ticket_pdf.pdf").pipe(file);
```

Output:

File downloaded successfully as downloaded_ticket.pdf

DELETING A FILE Query:

```
var fileId = db.fs.files.findOne({filename: "ticket_pdf.pdf"})._id;
db.fs.chunks.deleteMany({files_id: fileId});
db.fs.files.deleteOne({_id: fileId});
```

Output:

File ticket_pdf.pdf deleted from GridFS.

Class Performance	
Record	
Viva	
Total	

RESULT :

Thus, the query for GridFS for the Task Management System has been executed successfully.

```
D:\DHIVAKARAN>nslookup
Default Server:  dns.google
Address:  8.8.8.8

> www.google.com
Server:  dns.google
Address:  8.8.8.8

DNS request timed out.
    timeout was 2 seconds.
Name:    www.google.com
Address:  2404:6800:4007:803::2004
```

```
D:\DHIVAKARAN>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49684        LAPTOP-QHUAHIF8:49685  ESTABLISHED
  TCP    127.0.0.1:49685        LAPTOP-QHUAHIF8:49684  ESTABLISHED
  TCP    127.0.0.1:49686        LAPTOP-QHUAHIF8:49687  ESTABLISHED
  TCP    127.0.0.1:49687        LAPTOP-QHUAHIF8:49686  ESTABLISHED
  TCP    127.0.0.1:49697        LAPTOP-QHUAHIF8:49698  ESTABLISHED
  TCP    127.0.0.1:49698        LAPTOP-QHUAHIF8:49697  ESTABLISHED
  TCP    127.0.0.1:49699        LAPTOP-QHUAHIF8:49700  ESTABLISHED
  TCP    127.0.0.1:49700        LAPTOP-QHUAHIF8:49699  ESTABLISHED
  TCP    192.192.100.166:49418  20.198.162.78:https    ESTABLISHED
  TCP    192.192.100.166:52128  ec2-54-169-7-73:http   ESTABLISHED
  TCP    192.192.100.166:52140  a23-215-7-19:https     ESTABLISHED
  TCP    192.192.100.166:52143  a23-58-31-18:http      ESTABLISHED
  TCP    192.192.100.166:52145  52.98.87.242:https     ESTABLISHED
  TCP    192.192.100.166:52232  ec2-54-169-7-73:http   ESTABLISHED
  TCP    192.192.100.166:52326  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52327  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52328  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52329  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52330  a23-11-215-147:https   ESTABLISHED
  TCP    192.192.100.166:52331  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52423  whatsapp-cdn-shv-01-maa2:https  CLOSE_WAIT
  TCP    192.192.100.166:52424  whatsapp-chatd-edge-shv-03-sin6:5222  ESTABLISHED
  TCP    192.192.100.166:52558  52.167.17.97:https     FIN_WAIT_1
  TCP    192.192.100.166:52561  a184-87-193-160:https  ESTABLISHED
  TCP    192.192.100.166:52562  a184-87-193-160:https  ESTABLISHED
  TCP    192.192.100.166:52563  a184-87-193-160:https  ESTABLISHED
  TCP    192.192.100.166:52576  20.189.173.15:https    ESTABLISHED
  TCP    192.192.100.166:52581  a184-87-193-160:https  ESTABLISHED
  TCP    192.192.100.166:52587  ec2-18-138-86-144:http  TIME_WAIT
  TCP    192.192.100.166:52588  a23-46-187-176:https   ESTABLISHED
  TCP    192.192.100.166:52590  13.95.31.18:https      SYN_SENT
```

```
D:\DHIVAKARAN>tracert google.com

Tracing route to google.com [142.250.182.238]
over a maximum of 30 hops:

  1    60 ms    10 ms    86 ms  www.ltu.edu.tw [192.192.100.1]
  2    40 ms   111 ms   225 ms  14.102.13.129
  3    28 ms    66 ms     *     61.14.228.85
  4   157 ms   258 ms   421 ms  103.228.174.11
  5    66 ms    77 ms   297 ms  142.250.209.75
  6   176 ms   116 ms   446 ms  142.250.62.66
  7     *      657 ms    80 ms  72.14.232.34
  8  1068 ms   279 ms   104 ms  142.250.208.227
  9     *     2233 ms     *     142.250.214.105
 10     *      970 ms  1284 ms  bom07s29-in-f14.1e100.net [142.250.182.238]

Trace complete.
```

```
D:\DHIVAKARAN>netstat -p

Active Connections

  Proto  Local Address          Foreign Address        State

D:\DHIVAKARAN>netstat -o

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    127.0.0.1:49684        LAPTOP-QHUAHIF8:49685  ESTABLISHED     6816
  TCP    127.0.0.1:49685        LAPTOP-QHUAHIF8:49684  ESTABLISHED     6816
  TCP    127.0.0.1:49686        LAPTOP-QHUAHIF8:49687  ESTABLISHED     6816
  TCP    127.0.0.1:49687        LAPTOP-QHUAHIF8:49686  ESTABLISHED     6816
  TCP    127.0.0.1:49697        LAPTOP-QHUAHIF8:49698  ESTABLISHED     1280
  TCP    127.0.0.1:49698        LAPTOP-QHUAHIF8:49697  ESTABLISHED     1280
  TCP    127.0.0.1:49699        LAPTOP-QHUAHIF8:49700  ESTABLISHED     2756
  TCP    127.0.0.1:49700        LAPTOP-QHUAHIF8:49699  ESTABLISHED     2756
  TCP    192.192.100.166:49418  20.198.162.78:https    ESTABLISHED     6212
  TCP    192.192.100.166:52128  ec2-54-169-7-73:http   ESTABLISHED     4100
  TCP    192.192.100.166:52140  a23-215-7-19:https     ESTABLISHED     21456
  TCP    192.192.100.166:52143  a23-58-31-18:http      ESTABLISHED     21456
  TCP    192.192.100.166:52145  52.98.87.242:https     ESTABLISHED     21456
  TCP    192.192.100.166:52232  ec2-54-169-7-73:http   ESTABLISHED     22088
  TCP    192.192.100.166:52326  a23-11-215-147:https   CLOSE_WAIT      21456
  TCP    192.192.100.166:52327  a23-11-215-147:https   CLOSE_WAIT      21456
  TCP    192.192.100.166:52328  a23-11-215-147:https   CLOSE_WAIT      21456
  TCP    192.192.100.166:52329  a23-11-215-147:https   CLOSE_WAIT      21456
  TCP    192.192.100.166:52330  a23-11-215-147:https   ESTABLISHED     21456
  TCP    192.192.100.166:52331  a23-11-215-147:https   CLOSE_WAIT      21456
  TCP    192.192.100.166:52423  whatsapp-cdn-shv-01-maa2:https  CLOSE_WAIT      12244
  TCP    192.192.100.166:52424  whatsapp-chatd-edge-shv-03-sin6:5222  ESTABLISHED
  TCP    192.192.100.166:52576  20.189.173.15:https    ESTABLISHED     5804
  TCP    192.192.100.166:52591  13.95.31.18:https      TIME_WAIT       0
  TCP    192.192.100.166:52592  a23-46-230-144:http    LAST_ACK        1984
```

```
D:\DHIVAKARAN>netstat -h

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-i] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

  -a            Displays all connections and listening ports.
  -b            Displays the executable involved in creating each connection or
                listening port. In some cases well-known executables host
                multiple independent components, and in these cases the
                sequence of components involved in creating the connection
                or listening port is displayed. In this case the executable
                name is in [] at the bottom, on top is the component it called,
                and so forth until TCP/IP was reached. Note that this option
                can be time-consuming and will fail unless you have sufficient
                permissions.
  -e            Displays Ethernet statistics. This may be combined with the -s
                option.
  -f            Displays Fully Qualified Domain Names (FQDN) for foreign
                addresses.
  -i            Displays the time spent by a TCP connection in its current state.
  -n            Displays addresses and port numbers in numerical form.
  -o            Displays the owning process ID associated with each connection.
  -p proto      Shows connections for the protocol specified by proto; proto
                may be any of: TCP, UDP, TCPv6, or UDPv6.  If used with the -s
                option to display per-protocol statistics, proto may be any of:
                IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
  -q            Displays all connections, listening ports, and bound
                nonlistening TCP ports. Bound nonlistening ports may or may not
                be associated with an active connection.
  -r            Displays the routing table.
  -s            Displays per-protocol statistics.  By default, statistics are
                shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;
                the -p option may be used to specify a subset of the default.
  -t            Displays the current connection offload state.
  -x            Displays NetworkDirect connections, listeners, and shared
                endpoints.
  -y            Displays the TCP connection template for all connections.
                Cannot be combined with the other options.
  interval      Redisplays selected statistics, pausing interval seconds
                between each display.  Press CTRL+C to stop redisplaying
                statistics.  If omitted, netstat will print the current
                configuration information once.
```

```
D:\DHIVAKARAN>netstat -e
Interface Statistics

                           Received            Sent

Bytes                    13590220         7533134
Unicast packets             20342           21336
Non-unicast packets         28301            6825
Discards                        0               0
Errors                          0               0
Unknown protocols               0
```

```
D:\DHIVAKARAN>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:445            LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:3306           LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:5040           LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:9955           LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:33060          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:49664          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:49665          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:49666          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:49667          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:49672          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    0.0.0.0:49690          LAPTOP-QHUAHIF8:0      LISTENING
  TCP    127.0.0.1:27017        LAPTOP-QHUAHIF8:0      LISTENING
  TCP    127.0.0.1:49684        LAPTOP-QHUAHIF8:49685  ESTABLISHED
  TCP    127.0.0.1:49685        LAPTOP-QHUAHIF8:49684  ESTABLISHED
  TCP    127.0.0.1:49686        LAPTOP-QHUAHIF8:49687  ESTABLISHED
  TCP    127.0.0.1:49687        LAPTOP-QHUAHIF8:49686  ESTABLISHED
  TCP    127.0.0.1:49697        LAPTOP-QHUAHIF8:49698  ESTABLISHED
  TCP    127.0.0.1:49698        LAPTOP-QHUAHIF8:49697  ESTABLISHED
  TCP    127.0.0.1:49699        LAPTOP-QHUAHIF8:49700  ESTABLISHED
  TCP    127.0.0.1:49700        LAPTOP-QHUAHIF8:49699  ESTABLISHED
  TCP    192.192.100.166:139    LAPTOP-QHUAHIF8:0      LISTENING
  TCP    192.192.100.166:49418  20.198.162.78:https    ESTABLISHED
  TCP    192.192.100.166:52128  ec2-54-169-7-73:http   ESTABLISHED
  TCP    192.192.100.166:52140  a23-215-7-19:https     ESTABLISHED
  TCP    192.192.100.166:52143  a23-58-31-18:http      ESTABLISHED
  TCP    192.192.100.166:52145  52.98.87.242:https     ESTABLISHED
  TCP    192.192.100.166:52232  ec2-54-169-7-73:http   ESTABLISHED
  TCP    192.192.100.166:52326  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52327  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52328  a23-11-215-147:https   CLOSE_WAIT
  TCP    192.192.100.166:52329  a23-11-215-147:https   CLOSE_WAIT
```

```
D:\DHIVAKARAN>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
```

```
D:\DHIVAKARAN>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : LAPTOP-QHUAHIF8
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Mixed
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   Physical Address. . . . . . . . . : 2C-7B-A0-76-8E-29
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Wi-Fi 6E AX211 160MHz
   Physical Address. . . . . . . . . : 2C-7B-A0-76-8E-28
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek Gaming GbE Family Controller
   Physical Address. . . . . . . . . : 28-C5-C8-47-C5-F5
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
```

```
C:\Windows\System32>netstat -b

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49684        LAPTOP-QHUAHIF8:49685  ESTABLISHED
 [mysqld.exe]
  TCP    127.0.0.1:49685        LAPTOP-QHUAHIF8:49684  ESTABLISHED
 [mysqld.exe]
  TCP    127.0.0.1:49686        LAPTOP-QHUAHIF8:49687  ESTABLISHED
 [mysqld.exe]
  TCP    127.0.0.1:49687        LAPTOP-QHUAHIF8:49686  ESTABLISHED
 [mysqld.exe]
  TCP    127.0.0.1:49697        LAPTOP-QHUAHIF8:49698  ESTABLISHED
 [WUDFHost.exe]
  TCP    127.0.0.1:49698        LAPTOP-QHUAHIF8:49697  ESTABLISHED
 [WUDFHost.exe]
  TCP    127.0.0.1:49699        LAPTOP-QHUAHIF8:49700  ESTABLISHED
 [NVDisplay.Container.exe]
  TCP    127.0.0.1:49700        LAPTOP-QHUAHIF8:49699  ESTABLISHED
 [NVDisplay.Container.exe]
  TCP    192.168.106.115:49420  4.213.25.242:https     ESTABLISHED
  WpnService
 [svchost.exe]
  TCP    192.168.106.115:49421  4.213.25.240:https     ESTABLISHED
  WpnService
 [svchost.exe]
  TCP    192.168.106.115:52664  13.107.213.254:https   CLOSE_WAIT
 [SearchHost.exe]
  TCP    192.168.106.115:52671  52.98.57.114:https     TIME_WAIT
  TCP    192.168.106.115:52681  20.54.232.160:https    TIME_WAIT
  TCP    192.168.106.115:52682  a23-11-215-147:https   ESTABLISHED
 [SearchHost.exe]
  TCP    192.168.106.115:52683  52.98.57.114:https     ESTABLISHED
 [SearchHost.exe]
  TCP    192.168.106.115:52684  52.98.57.114:https     ESTABLISHED
 [SearchHost.exe]
```

EXPT NO : 13	TASK MANAGER SYSTEM
DATE :	

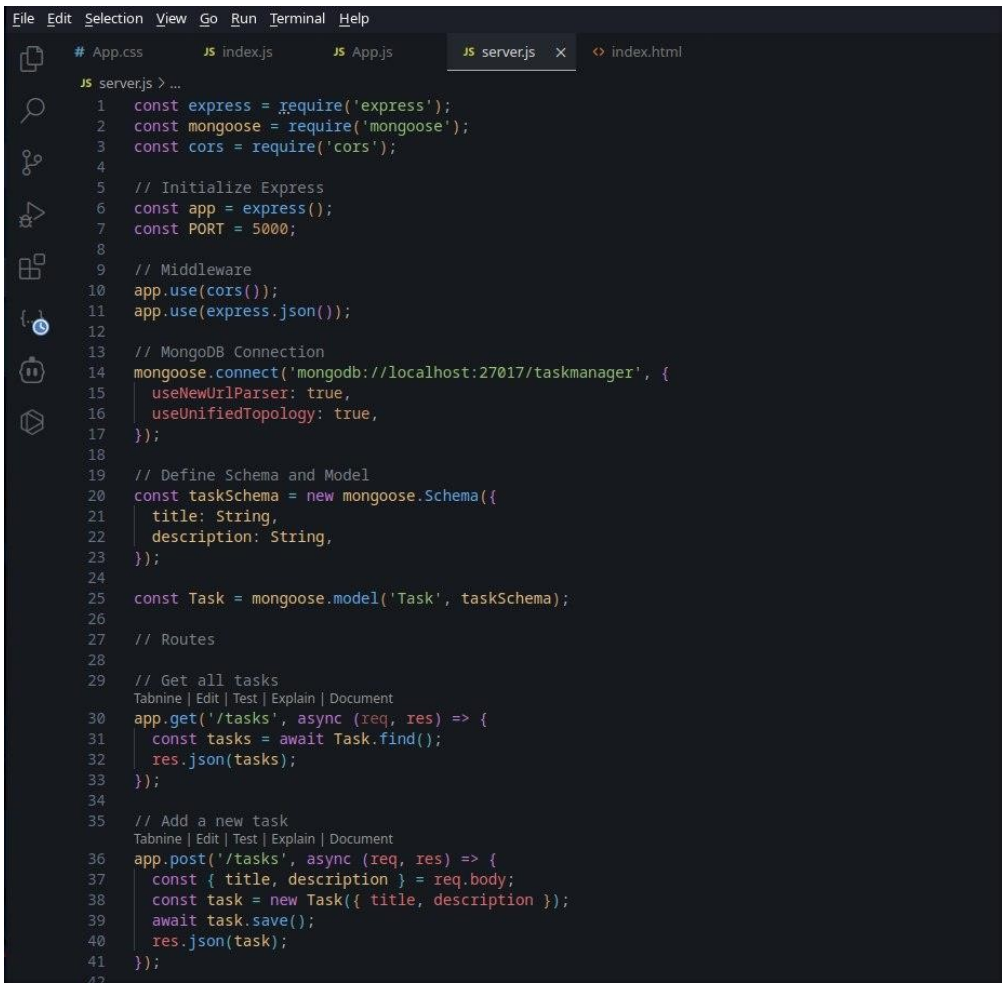
AIM:

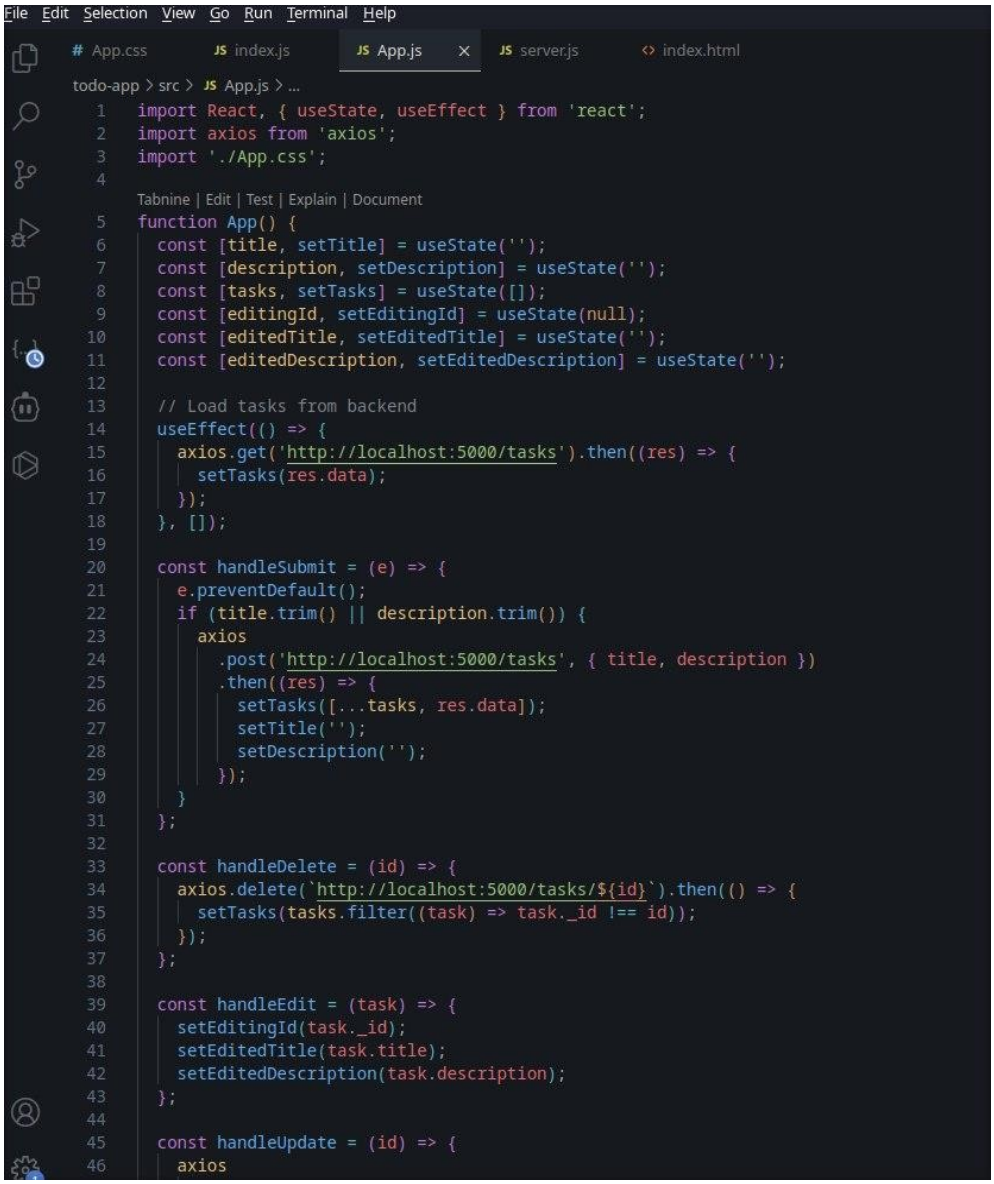
To implement the Task manager system using mongodb.

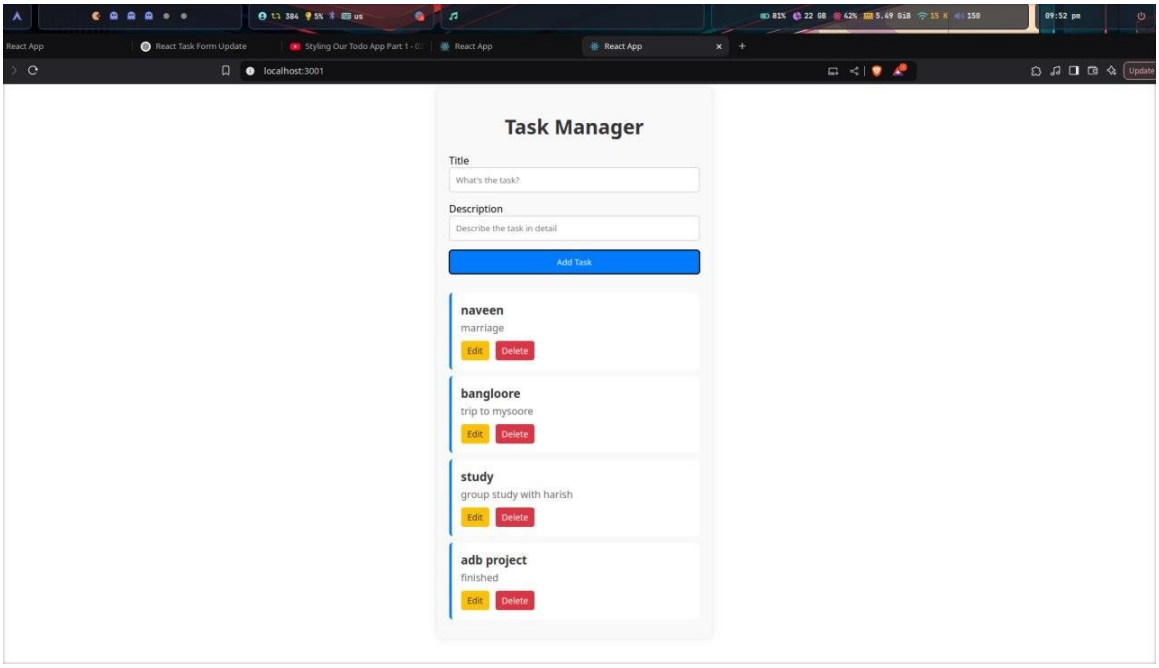
EXPLANATION:

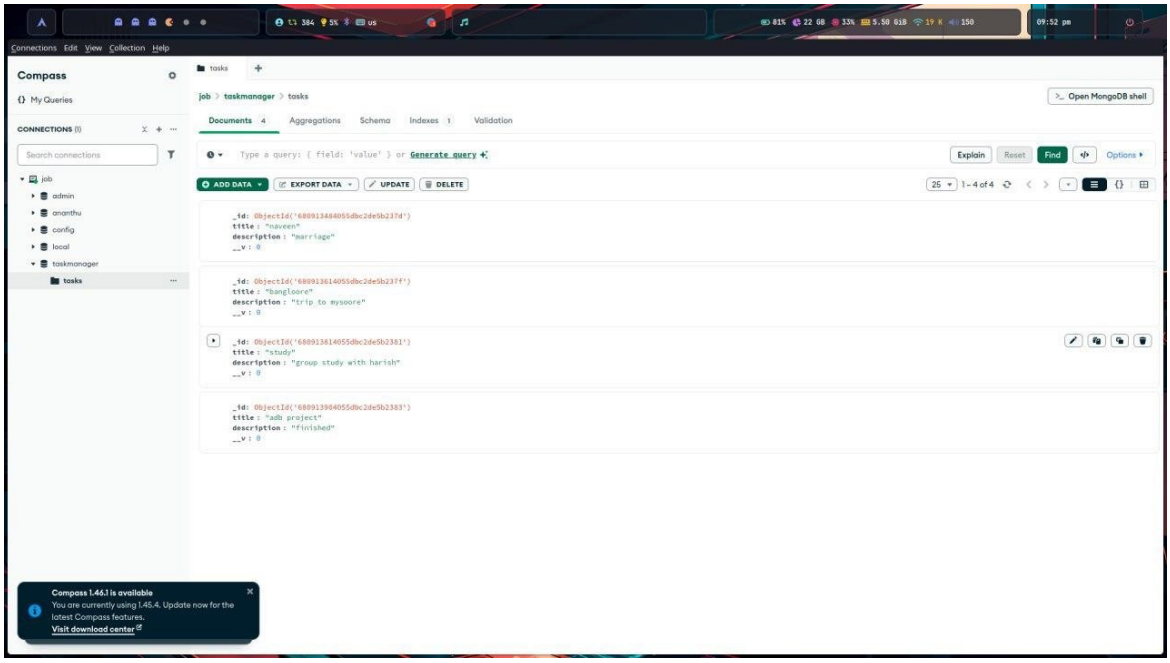
A **Task Manager System** is a software application designed to help users organize, track, and manage tasks efficieny

PROGRAM:




File Edit Selection View Go Run Terminal Help
App.css
index.js
App.js
server.js
index.html
todo-app > src > App.js > ...
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import './App.css';
Tabnine | Edit | Test | Explain | Document
function App() {
const [title, setTitle] = useState('');
const [description, setDescription] = useState('');
const [tasks, setTasks] = useState([]);
const [editingId, setEditingId] = useState(null);
const [editedTitle, setEditedTitle] = useState('');
const [editedDescription, setEditedDescription] = useState('');
// Load tasks from backend
useEffect(() => {
axios.get('http://localhost:5000/tasks').then((res) => {
setTasks(res.data);
});
}, []);
const handleSubmit = (e) => {
e.preventDefault();
if (title.trim() || description.trim()) {
axios
.post('http://localhost:5000/tasks', { title, description })
.then((res) => {
setTasks([...tasks, res.data]);
setTitle('');
setDescription('');
});
}
};
const handleDelete = (id) => {
axios.delete(`http://localhost:5000/tasks/${id}`).then(() => {
setTasks(tasks.filter((task) => task._id !== id));
});
};
const handleEdit = (task) => {
setEditingId(task._id);
setEditedTitle(task.title);
setEditedDescription(task.description);
};
const handleUpdate = (id) => {
axios


localhost:3001
Task Manager
Title
What's the task?
Description
Describe the task in detail
Add Task
naveen
marriage
Edit
Delete
bangloore
trip to mysoore
Edit
Delete
study
group study with harish
Edit
Delete
adb project
finished
Edit
Delete


Compass
My Queries
CONNECTIONS (1)
Search connections
job
admin
ananthu
config
local
taskmanager
tasks
job > taskmanager > tasks
Open MongoDB shell
Documents 4
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
1 - 4 of 4
_id: ObjectId('680913484055dbc2de5b237d')
title : "naveen"
description : "marriage"
__v : 0
_id: ObjectId('680913614055dbc2de5b237f')
title : "bangloore"
description : "trip to mysoore"
__v : 0
_id: ObjectId('680913814055dbc2de5b2381')
title : "study"
description : "group study with harish"
__v : 0
_id: ObjectId('680913904055dbc2de5b2383')
title : "adb project"
description : "finished"
__v : 0
Compass 1.46.1 is available
You are currently using 1.45.4. Update now for the latest Compass features.
Visit download center

CLASS PERFORMANCE	
VIVA	
RECORD	
TOTAL	

RESULT:

Thus,the task manager system implemented successfully

SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

Affiliated to Anna University, Chennai

Re-Accredited by NAAC with "A", Recognized by UGC with Section 2(f) and 12(B)

NBA Accredited UG Programmes : Agri, BME, BT, CSE, ECE, EEE, MECH and IT Coimbatore - 641 062,

L & T By Pass, Tamil Nadu, India

POWERING THE YOUTH
EMPOWERING THE NATION

21CS421 – ADVANCED DATABASES LABORATORY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)
Affiliated to Anna University, Chennai
Re-Accredited by NAAC with "A", Recognized by UGC with Section 2(f) and 12(B)
NBA Accredited UG Programmes : Agri, BME, BT, CSE, ECE, EEE, MECH and IT Coimbatore - 641 062,
L & T By Pass, Tamil Nadu, India

21CS421 – ADVANCED DATABASES LABORATORY

RECORD

NAME: _____ ROLLNO: _____

CLASS: _____ BRANCH: _____

ACADEMIC YEAR: 2024 - 2025 BATCH: 2023 – 2027 SEMESTER: 4

Certified and bonafide record of work done by ……..…….……

Place:

Date:

Staff In-Charge Head of the Department

University Register Number: ……..

Submitted for the University Practical Examination held on ……..……

INTERNAL EXAMINER EXTERNAL EXAMINER

EX NO	DATE	NAME OF THE EXPERIMENT	MARKS
1		CASE STUDY OF SQL VS NON SQL AND INSTALLATION OF MONGODB	
2		BASIC CRUD OPERATIONS 1 [INSERT]	
3		BASIC CRUD OPERATIONS II [UPDATE,REMOVE] & UPSERT	
4		QUERY CONDITIONALS	
5		QUERYING ON ARRAYS , EMBEDDED DOCUMENTS & TYPE SPECIFIC QUERIES	
6		QUERYING ON CAPPED COLLECTIONS	
7		QUERYING USING CURSORS	
8		QUERYING ON INDEXING & COMPOUND INDEXES	
9		GEOSPATIAL INDEXES IN MONGODB	
10		AGGREGATION AND PIPELINE OPERATIONS	
11		MAP REDUCE AND REPLICATION	
12		GRIDFS USING MONGODB	
13		TASK MANAGEMENT SYSTEM	
		AVERAGE:	

SINGNATURE OF THE FACULTY