

### Database Requirements:

- Create a database named ToDoDB.  
Use ToDoDb
- Design a collection named tasks.  
db.createCollection('tasks')
- Each task should have information like title, description, status (e.g., "Pending," "Completed"), and due date.

### CRUD Operations:

#### Create Operation:

- Insert at least 5 tasks with various titles, descriptions, statuses, and due dates.

```
db.tasks.insertMany([
  {
    title: "Finish Report",
    description: "Write a detailed report on the quarterly financial
performance.",
    status: "Pending",
    dueDate: ISODate("2024-03-15")
  },
  {
    title: "Prepare Presentation",
    description: "Create a PowerPoint presentation for the
upcoming meeting.",
    status: "Completed",
    dueDate: ISODate("2024-03-20")
  },
  {
    title: "Review Code",
    description: "Review and provide feedback on the latest code
changes.",
    status: "Pending",
    dueDate: ISODate("2024-03-10")
  },
  {
    title: "Client Meeting",
```

```

        description: "Schedule and prepare for the client meeting.",
        status: "Pending",
        dueDate: ISODate("2024-03-25")
    },
    {
        title: "Project Kickoff",
        description: "Plan and organize the project kickoff event.",
        status: "Pending",
        dueDate: ISODate("2024-03-05")
    }
]);

```

### Read Operation:

- Retrieve and display all tasks in the database.  
db.tasks.find()
- Display the details of a specific task by searching with its title or status.  
db.tasks.find({title: 'Project Kickoff'})  
db.tasks.find({status : 'Pending'})
- List all tasks that are pending or completed.  
db.tasks.find({ \$or : [ { status : { \$eq : 'Pending'}},  
{status : { \$eq : 'Completed'}}]})
- List of all task which is still pending after due date.  
db.tasks.find({\$expr :  
  {\$cond : {  
    if : { \$and : [{ \$eq :["\$status","Pending"]}, { \$lt: ["\$dueDate",new  
    Date()]]} } ,  
    then : true,  
    else: false  
  }}})

- Show all task in pending key and completed key {pending: [task....], completed: [task...]}

```
db.tasks.aggregate([
  {
    $group: {
      _id: "$status",
      tasks: { $push: "$$ROOT" }
    }
  }
])
```

### Update Operation:

- Update the status of a specific task (e.g., from "Pending" to "Completed").  

```
db.tasks.updateOne({title: 'Finish Report'},
  { $set : { status : 'Completed'}}
```
- Update the due date of a task.  

```
db.tasks.updateOne({title: 'Project Kickoff'},
  { $set : { dueDate : ISODate("2024-03-07")}})
```
- Update all pending task which due date is less than 2 days from today  

```
db.tasks.updateMany ({ dueDate :
  { $lt : (new Date().getDate()+2)}},
  { $set : { dueDate : ISODate("2024-03-24")}})
```

### Delete Operation:

- Remove a task by its title or status.  

```
db.tasks.deleteOne({title : 'Project Kickoff'})
```
- Delete all completed tasks.  

```
db.tasks.deleteMany({status : 'Completed'})
```

