Mongo-DB PROJECT

Data sets (json) Used in this:

- students
- faculty
- courses
- enrollments
- Activities

➔ What is the Project about ? :

We have Loaded all the Datasets in MongoDB Database .

Here We have given the **Question** and we have to perform **Queries** and we will be providing the answers by doing in the Shell (Mongosh) and paste here the solution.

Tech Stack Used:

- MongoDB (NoSQL Database)
- MongoDB Query Language (MQL)
- Aggregation Pipeline, $lookup, $group
- Word for Report Project.

➔ Learnings

- Applied CRUD & Aggregation operations
- Used $lookup for joins between collections
- Filtered, grouped & analyzed data
- Worked with real-life datasets (students, faculty, courses)

Submitted By:

KANCHAN KAPRI Rag: No 1240258215 (69)

Submitted To:

Mr. Ankit Verma

(Faculty)

Submission Date: 15 / 10 / 2025

PROJECT

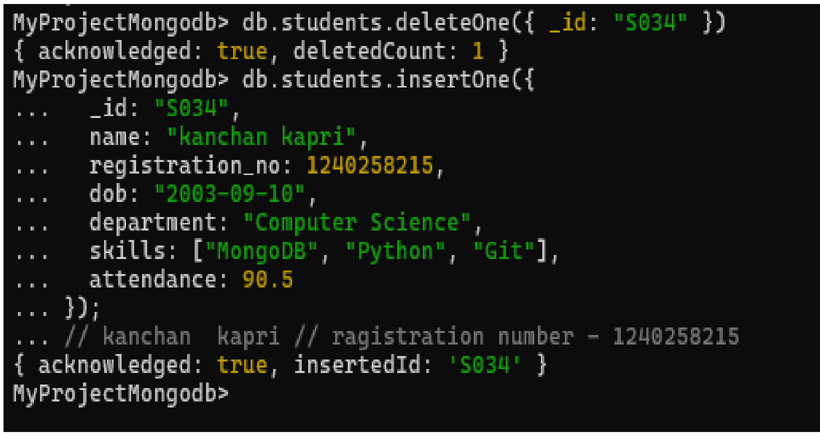
1. Complex Filters & Projections

Q1. List the names and departments of students who have more than 85% attendance and are skilled in both **"MongoDB"** and **"Python"**.

Solution:

Step 1: We do not Find Any document maybe no document was there which was matching to this question. But we created one record which has same requiornments then we had instead it to students.

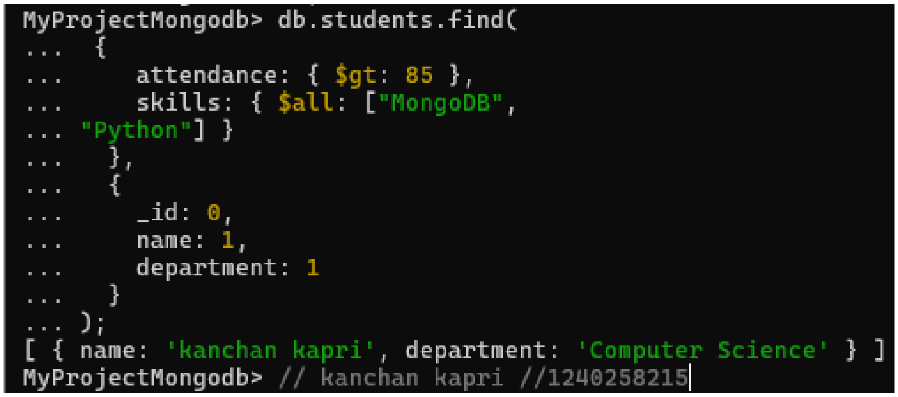
```
➔ db.students.deleteOne({ _id: "S034" })

➔ db.students.insertOne({
  ...  _id: "S034",
  ...  name: "kanchan kapri",
  ...  registration_no: 1240258215,
  ...  dob: "2003-09-10",
  ...  department: "Computer Science",
  ...  skills: ["MongoDB", "Python", "Git"],
  ...  attendance: 90.5
  ... });
```



```
MyProjectMongodb> db.students.deleteOne({ _id: "S034" })
{ acknowledged: true, deletedCount: 1 }
MyProjectMongodb> db.students.insertOne({
...   _id: "S034",
...   name: "kanchan kapri",
...   registration_no: 1240258215,
...   dob: "2003-09-10",
...   department: "Computer Science",
...   skills: ["MongoDB", "Python", "Git"],
...   attendance: 90.5
... });
... // kanchan  kapri // ragistration number - 1240258215
{ acknowledged: true, insertedId: 'S034' }
MyProjectMongodb>
```

Step 2: So now it has given the Output , and resulting student is same that we have creted as my Name.

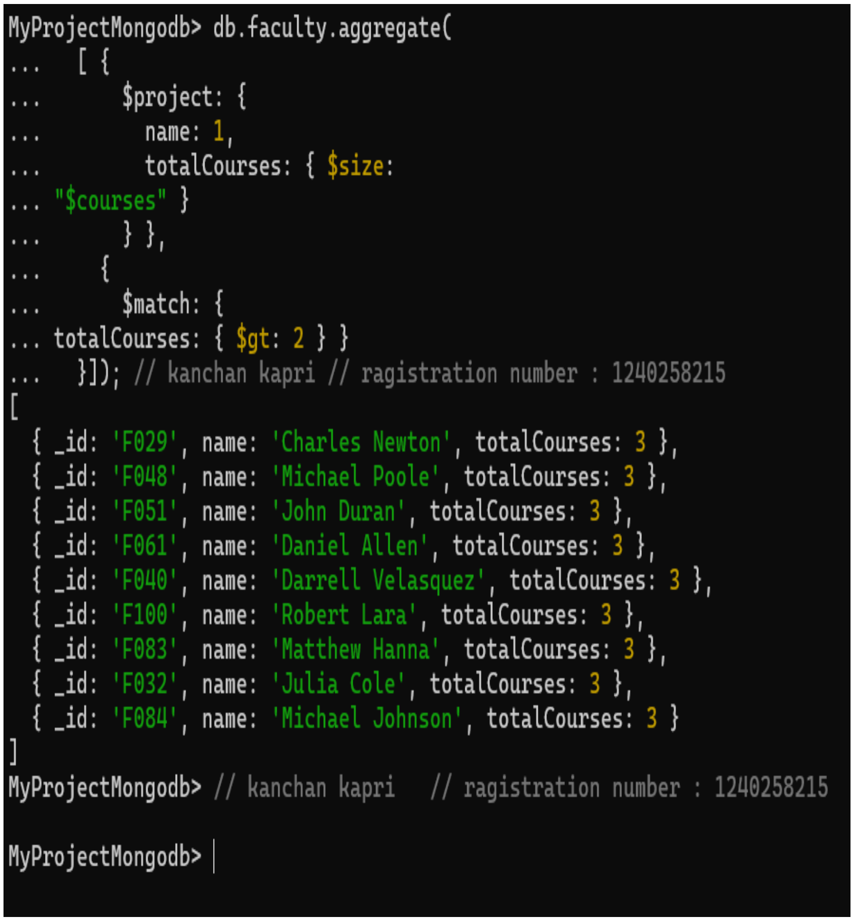
```
➔ db.students.find(
  ... {
  ...    attendance: { $gt: 85 },
  ...    skills: { $all: ["MongoDB",
  ... "Python"] }
  ...  },
  ...  {
  ...    _id: 0,
  ...    name: 1,
  ...    department: 1
  ...  }
  ... );
  [ { name: 'kanchan kapri', department: 'Computer Science' } ]
```



```
MyProjectMongodb> db.students.find(
...   {
...     attendance: { $gt: 85 },
...     skills: { $all: ["MongoDB",
... "Python"] }
...    },
...    {
...     _id: 0,
...     name: 1,
...     department: 1
...    }
... );
[ { name: 'kanchan kapri', department: 'Computer Science' } ]
MyProjectMongodb> // kanchan kapri //1240258215
```

Q2. Show all faculty who are teaching more than 2 courses. Display their names and the total number of courses they teach.

```
db.faculty.aggregate(                    // question 2 ....      //

...   [ {

...       $project: {

...         name: 1,

...         totalCourses: { $size:

... "$courses" }

...       } },

...     {

...       $match: {

... totalCourses: { $gt: 2 } }

...   }]); db.faculty.aggregate(

...   [ {

...       $project: {

...         name: 1,

...         totalCourses: { $size:

... "$courses" }

...       } },

...     {

...       $match: {

... totalCourses: { $gt: 2 } }

...   }]);
```

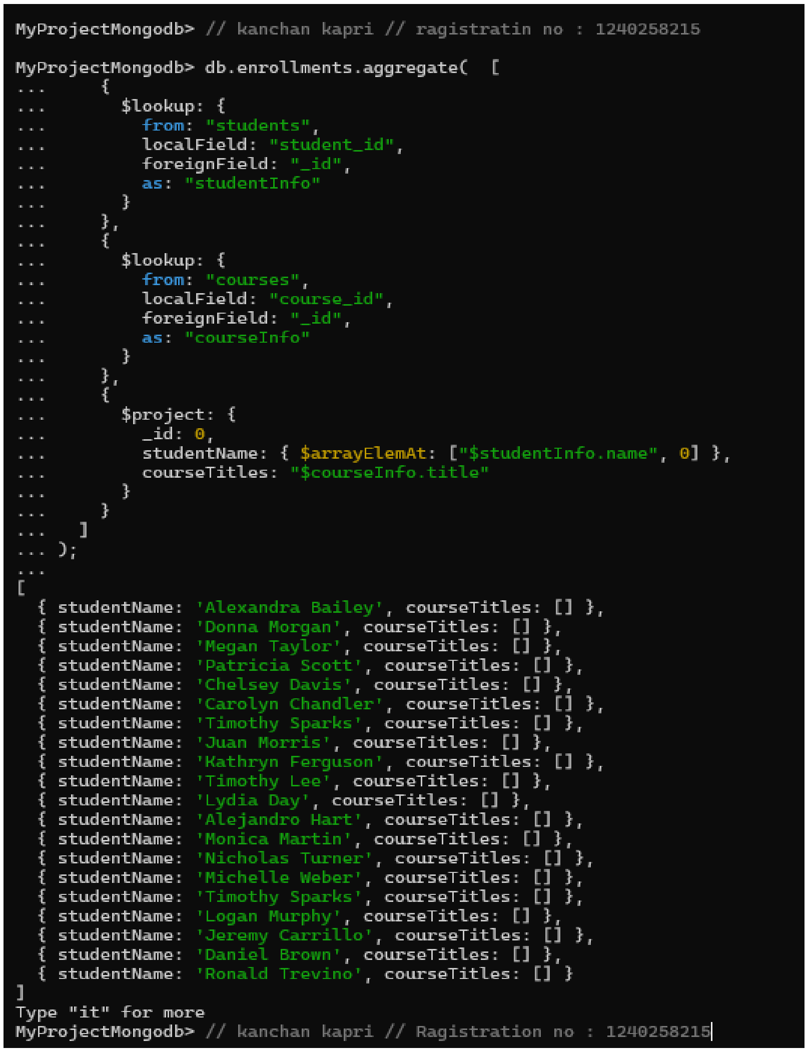


```
MyProjectMongodb> db.faculty.aggregate(
...   [ {
...       $project: {
...         name: 1,
...         totalCourses: { $size:
... "$courses" }
...       } },
...     {
...       $match: {
... totalCourses: { $gt: 2 } }
...   }]); // kanchan kapri // ragistration number : 1240258215
[
  { _id: 'F029', name: 'Charles Newton', totalCourses: 3 },
  { _id: 'F048', name: 'Michael Poole', totalCourses: 3 },
  { _id: 'F051', name: 'John Duran', totalCourses: 3 },
  { _id: 'F061', name: 'Daniel Allen', totalCourses: 3 },
  { _id: 'F040', name: 'Darrell Velasquez', totalCourses: 3 },
  { _id: 'F100', name: 'Robert Lara', totalCourses: 3 },
  { _id: 'F083', name: 'Matthew Hanna', totalCourses: 3 },
  { _id: 'F032', name: 'Julia Cole', totalCourses: 3 },
  { _id: 'F084', name: 'Michael Johnson', totalCourses: 3 }
]
MyProjectMongodb> // kanchan kapri   // ragistration number : 1240258215

MyProjectMongodb> |
```

2. Joins ($lookup) and Aggregations

Q3. Show each student’s name along with the course titles they are enrolled in (use $lookup between enrollments, students, and courses).

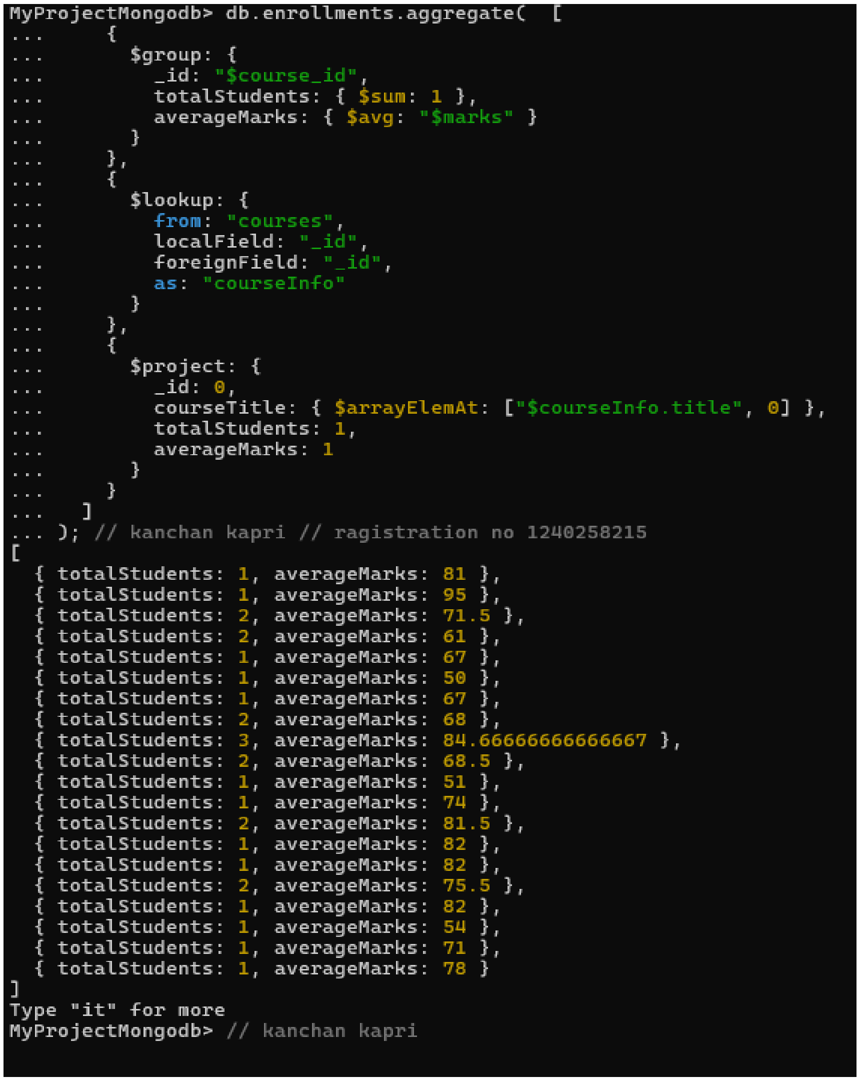
```
db.enrollments.aggregate(  [
...     {
...       $lookup: {
...         from: "students",
...         localField: "student_id",
...         foreignField: "_id",
...         as: "studentInfo"
...       }
...     },
...     {
...       $lookup: {
...         from: "courses",
...         localField: "course_id",
...         foreignField: "_id",
...         as: "courseInfo"
...       }
...     },
...     {
...       $project: {
...         _id: 0,
...         studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
...         courseTitles: "$courseInfo.title"
...       }
...     }
...   ]
... );
```



```
MyProjectMongodb> // kanchan kapri // ragistratin no : 1240258215

MyProjectMongodb> db.enrollments.aggregate(  [
...       {
...         $lookup: {
...           from: "students",
...           localField: "student_id",
...           foreignField: "_id",
...           as: "studentInfo"
...         }
...       },
...       {
...         $lookup: {
...           from: "courses",
...           localField: "course_id",
...           foreignField: "_id",
...           as: "courseInfo"
...         }
...       },
...       {
...         $project: {
...           _id: 0,
...           studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
...           courseTitles: "$courseInfo.title"
...         }
...       }
...     ]
... );
...
[
  { studentName: 'Alexandra Bailey', courseTitles: [] },
  { studentName: 'Donna Morgan', courseTitles: [] },
  { studentName: 'Megan Taylor', courseTitles: [] },
  { studentName: 'Patricia Scott', courseTitles: [] },
  { studentName: 'Chelsey Davis', courseTitles: [] },
  { studentName: 'Carolyn Chandler', courseTitles: [] },
  { studentName: 'Timothy Sparks', courseTitles: [] },
  { studentName: 'Juan Morris', courseTitles: [] },
  { studentName: 'Kathryn Ferguson', courseTitles: [] },
  { studentName: 'Timothy Lee', courseTitles: [] },
  { studentName: 'Lydia Day', courseTitles: [] },
  { studentName: 'Alejandro Hart', courseTitles: [] },
  { studentName: 'Monica Martin', courseTitles: [] },
  { studentName: 'Nicholas Turner', courseTitles: [] },
  { studentName: 'Michelle Weber', courseTitles: [] },
  { studentName: 'Timothy Sparks', courseTitles: [] },
  { studentName: 'Logan Murphy', courseTitles: [] },
  { studentName: 'Jeremy Carrillo', courseTitles: [] },
  { studentName: 'Daniel Brown', courseTitles: [] },
  { studentName: 'Ronald Trevino', courseTitles: [] }
]
Type "it" for more
MyProjectMongodb> // kanchan kapri // Ragistration no : 1240258215
```

Q4. For each course, display the course title, number of students enrolled, and average marks (use $group).

```
db.enrollments.aggregate(  [

...     {

...       $group: {

...         _id: "$course_id",

...         totalStudents: { $sum: 1 },

...         averageMarks: { $avg: "$marks" }

...       }

...     },

...     {

...       $lookup: {

...         from: "courses",

...         localField: "_id",

...         foreignField: "_id",

...         as: "courseInfo"

...       }

...     },

...     {

...       $project: {

...         _id: 0,

...         courseTitle: { $arrayElemAt: ["$courseInfo.title", 0] },

...         totalStudents: 1,

...         averageMarks: 1

...       }

...     }

...   ]

... );
```

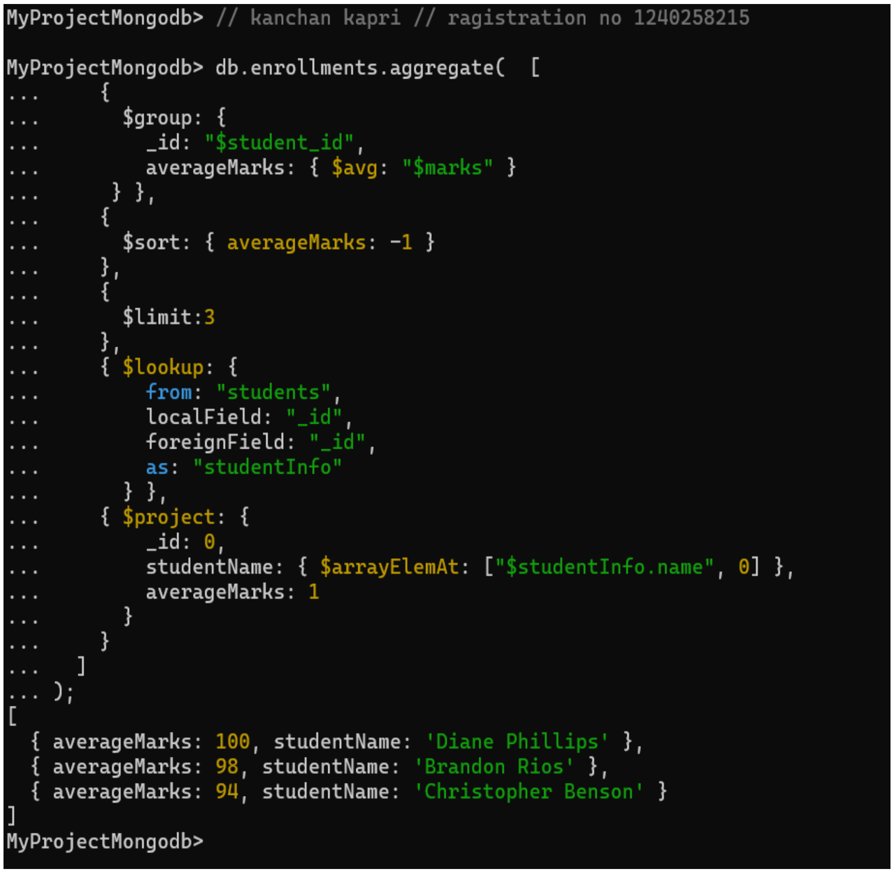


```
MyProjectMongodb> db.enrollments.aggregate(  [
...       {
...         $group: {
...           _id: "$course_id",
...           totalStudents: { $sum: 1 },
...           averageMarks: { $avg: "$marks" }
...         }
...       },
...       {
...         $lookup: {
...           from: "courses",
...           localField: "_id",
...           foreignField: "_id",
...           as: "courseInfo"
...         }
...       },
...       {
...         $project: {
...           _id: 0,
...           courseTitle: { $arrayElemAt: ["$courseInfo.title", 0] },
...           totalStudents: 1,
...           averageMarks: 1
...         }
...       }
...     ]
... ); // kanchan kapri // ragistration no 1240258215
[
  { totalStudents: 1, averageMarks: 81 },
  { totalStudents: 1, averageMarks: 95 },
  { totalStudents: 2, averageMarks: 71.5 },
  { totalStudents: 2, averageMarks: 61 },
  { totalStudents: 1, averageMarks: 67 },
  { totalStudents: 1, averageMarks: 50 },
  { totalStudents: 1, averageMarks: 67 },
  { totalStudents: 2, averageMarks: 68 },
  { totalStudents: 3, averageMarks: 84.66666666666667 },
  { totalStudents: 2, averageMarks: 68.5 },
  { totalStudents: 1, averageMarks: 51 },
  { totalStudents: 1, averageMarks: 74 },
  { totalStudents: 2, averageMarks: 81.5 },
  { totalStudents: 1, averageMarks: 82 },
  { totalStudents: 1, averageMarks: 82 },
  { totalStudents: 2, averageMarks: 75.5 },
  { totalStudents: 1, averageMarks: 82 },
  { totalStudents: 1, averageMarks: 54 },
  { totalStudents: 1, averageMarks: 71 },
  { totalStudents: 1, averageMarks: 78 }
]
Type "it" for more
MyProjectMongodb> // kanchan kapri
```

3. Grouping, Sorting, and Limiting

Q5. Find the top 3 students with the highest average marks across all enrolled courses.

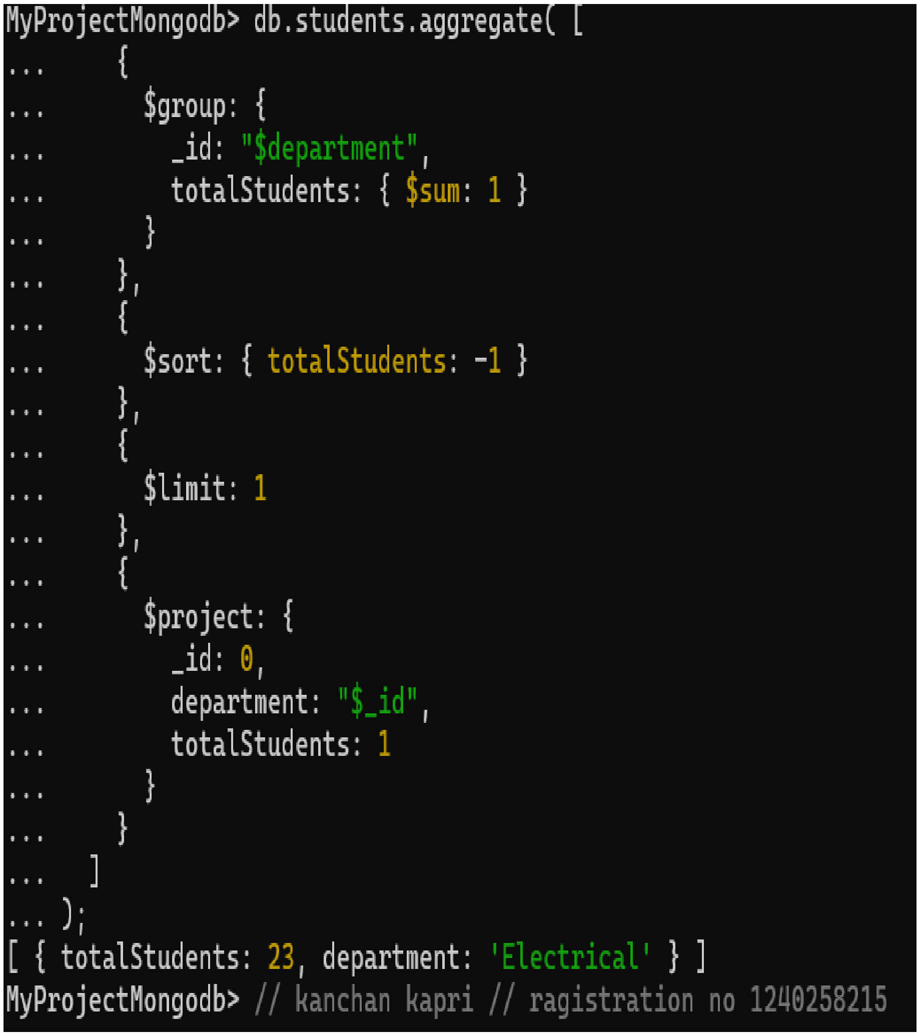
```
db.enrollments.aggregate(  [

...    {

...     $group: {

...       _id: "$student_id",

...       averageMarks: { $avg: "$marks" }

...     } },

...    {

...     $sort: { averageMarks: -1 }

...    },

...    {

...     $limit:3

...    },

...    { $lookup: {

...       from: "students",

...       localField: "_id",

...       foreignField: "_id",

...       as: "studentInfo"

...     } },

...    { $project: {

...       _id: 0,

...       studentName: { $arrayElemAt: ["$studentInfo.name", 0] },

...       averageMarks: 1

...     }

...    }

...  ]

... );
```



```
MyProjectMongodb> // kanchan kapri // ragistration no 1240258215

MyProjectMongodb> db.enrollments.aggregate(  [
...       {
...         $group: {
...           _id: "$student_id",
...           averageMarks: { $avg: "$marks" }
...         } },
...       {
...         $sort: { averageMarks: -1 }
...       },
...       {
...         $limit:3
...       },
...       { $lookup: {
...           from: "students",
...           localField: "_id",
...           foreignField: "_id",
...           as: "studentInfo"
...         } },
...       { $project: {
...           _id: 0,
...           studentName: { $arrayElemAt: ["$studentInfo.name", 0] },
...           averageMarks: 1
...         }
...       }
...     ]
... );
[
  { averageMarks: 100, studentName: 'Diane Phillips' },
  { averageMarks: 98, studentName: 'Brandon Rios' },
  { averageMarks: 94, studentName: 'Christopher Benson' }
]
MyProjectMongodb>
```

Q6. Count how many students are in each department. Display the department with the highest number of students.

```
db.students.aggregate( [
...   {
...     $group: {
...       _id: "$department",
...       totalStudents: { $sum: 1 }
...     }
...   },
...   {
...     $sort: { totalStudents: -1 }
...   },
...   {
...     $limit: 1
...   },
...   {
...     $project: {
...       _id: 0,
...       department: "$_id",
...       totalStudents: 1
...     }
...   }
... ]
... );
```

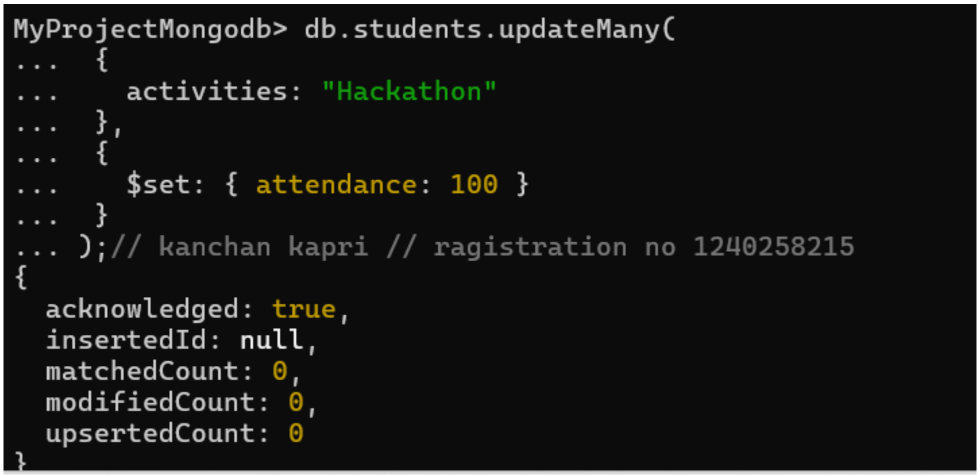


```
MyProjectMongodb> db.students.aggregate( [
...     {
...       $group: {
...         _id: "$department",
...         totalStudents: { $sum: 1 }
...       }
...     },
...     {
...       $sort: { totalStudents: -1 }
...     },
...     {
...       $limit: 1
...     },
...     {
...       $project: {
...         _id: 0,
...         department: "$_id",
...         totalStudents: 1
...       }
...     }
...   ]
... );
[ { totalStudents: 23, department: 'Electrical' } ]
MyProjectMongodb> // kanchan kapri // ragistration no 1240258215
```

4. Update, Upsert, and Delete

Q7. Update attendance to 100% for all students who won any "Hackathon".

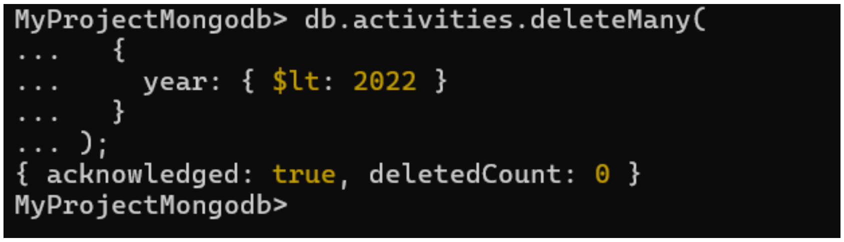
```
db.students.updateMany(
... {
...    activities: "Hackathon"
... },
... {
...    $set: { attendance: 100 }
... }
... );
```



```
MyProjectMongodb> db.students.updateMany(
...   {
...     activities: "Hackathon"
...   },
...   {
...     $set: { attendance: 100 }
...   }
... );// kanchan kapri // ragistration no 1240258215
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

Q8. Delete all student activity records where the activity year is before 2022.

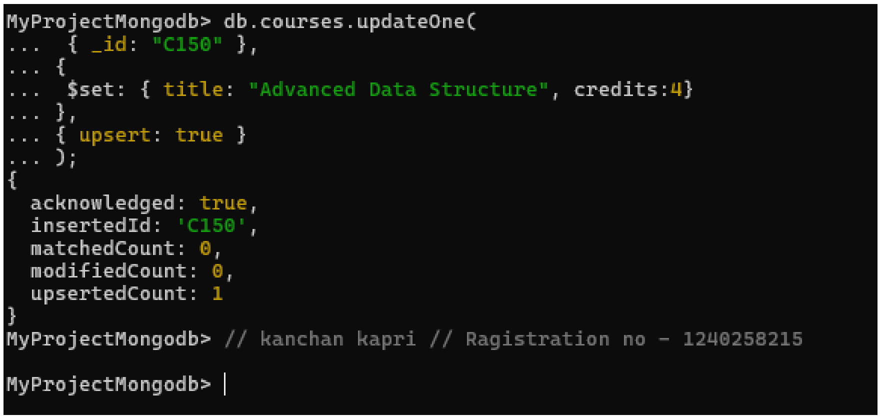
```
db.activities.deleteMany(

... {

... year: { $lt: 2022 }

... }

... );
```



```
MyProjectMongodb> db.activities.deleteMany(
...   {
...     year: { $lt: 2022 }
...   }
... );
{ acknowledged: true, deletedCount: 0 }
MyProjectMongodb>
```

Q9. Upsert a course record for "Data Structures" with ID "C150" and credits 4—if it doesn’t exist, insert it; otherwise update its title to "Advanced Data Structures".

```
db.courses.updateOne(

...   { _id: "C150" },

...   { $set: { title: "Advanced Data Structures", credits: 4 } },

...   { upsert: true }

... )

...

{

  acknowledged: true,

  insertedId: null,

  matchedCount: 1,

  modifiedCount: 1,

  upsertedCount: 0

}
```

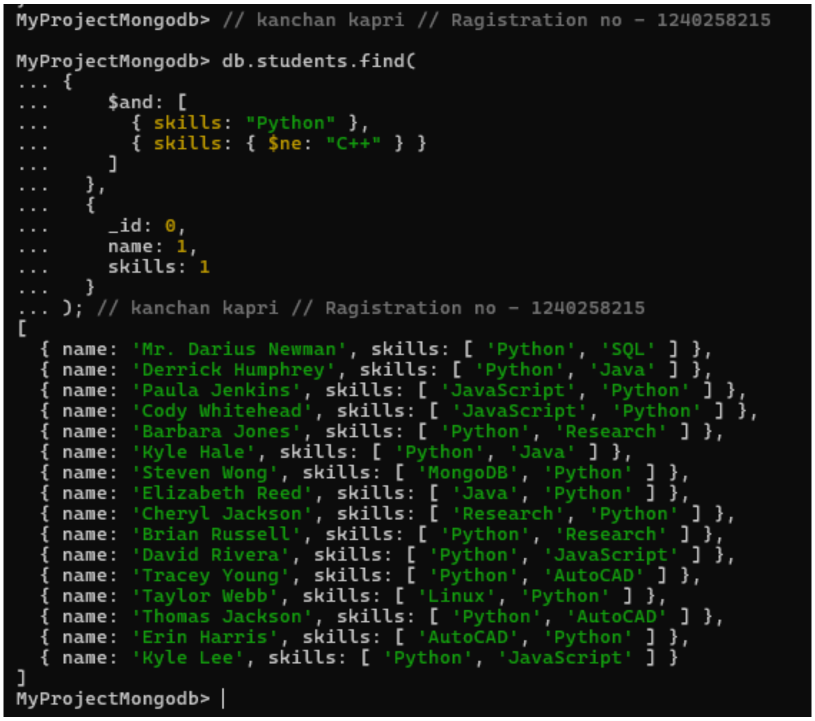


```
MyProjectMongodb> db.courses.updateOne(
...   { _id: "C150" },
... {
...   $set: { title: "Advanced Data Structure", credits:4}
... },
... { upsert: true }
... );
{
  acknowledged: true,
  insertedId: 'C150',
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
MyProjectMongodb> // kanchan kapri // Ragistration no - 1240258215

MyProjectMongodb> |
```

5. Array & Operator Usage

Q10. Find all students who have "Python" as a skill but not "C++".

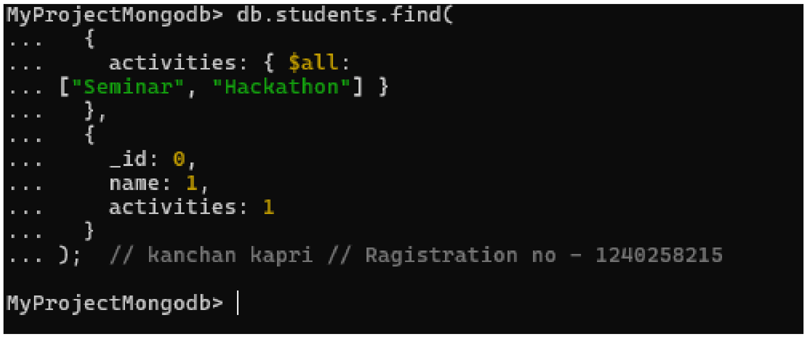
```
db.students.find(
... {
...    $and: [
...      { skills: "Python" },
...      { skills: { $ne: "C++" } }
...    ]
...  },
...  {
...    _id: 0,
...    name: 1,
...    skills: 1
...  }
... );
```



```
MyProjectMongodb> // kanchan kapri // Ragistration no - 1240258215

MyProjectMongodb> db.students.find(
... {
...     $and: [
...       { skills: "Python" },
...       { skills: { $ne: "C++" } }
...     ]
...   },
...   {
...     _id: 0,
...     name: 1,
...     skills: 1
...   }
... ); // kanchan kapri // Ragistration no - 1240258215
[
  { name: 'Mr. Darius Newman', skills: [ 'Python', 'SQL' ] },
  { name: 'Derrick Humphrey', skills: [ 'Python', 'Java' ] },
  { name: 'Paula Jenkins', skills: [ 'JavaScript', 'Python' ] },
  { name: 'Cody Whitehead', skills: [ 'JavaScript', 'Python' ] },
  { name: 'Barbara Jones', skills: [ 'Python', 'Research' ] },
  { name: 'Kyle Hale', skills: [ 'Python', 'Java' ] },
  { name: 'Steven Wong', skills: [ 'MongoDB', 'Python' ] },
  { name: 'Elizabeth Reed', skills: [ 'Java', 'Python' ] },
  { name: 'Cheryl Jackson', skills: [ 'Research', 'Python' ] },
  { name: 'Brian Russell', skills: [ 'Python', 'Research' ] },
  { name: 'David Rivera', skills: [ 'Python', 'JavaScript' ] },
  { name: 'Tracey Young', skills: [ 'Python', 'AutoCAD' ] },
  { name: 'Taylor Webb', skills: [ 'Linux', 'Python' ] },
  { name: 'Thomas Jackson', skills: [ 'Python', 'AutoCAD' ] },
  { name: 'Erin Harris', skills: [ 'AutoCAD', 'Python' ] },
  { name: 'Kyle Lee', skills: [ 'Python', 'JavaScript' ] }
]
MyProjectMongodb> |
```

Q11. Return names of students who participated in "Seminar" and "Hackathon" both.

```
db.students.find(
...  {
...    activities: { $all:
... ["Seminar", "Hackathon"] }
...  },
...  {
...    _id: 0,
...    name: 1,
...    activities: 1
...  }
... );
```

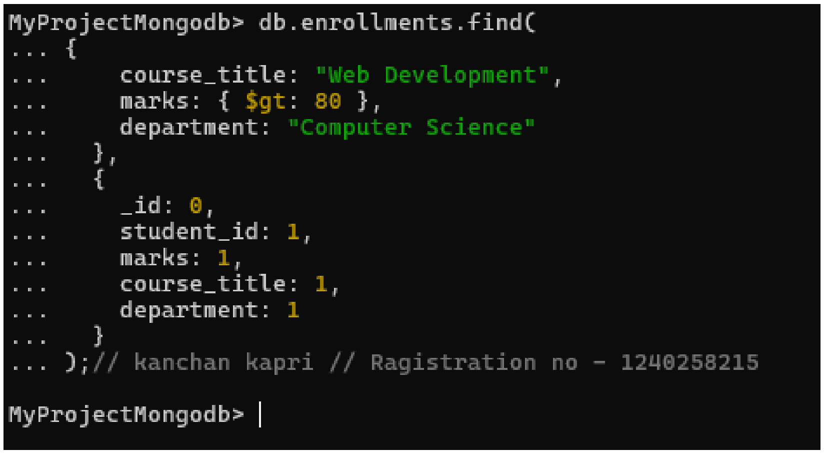


```
MyProjectMongodb> db.students.find(
...    {
...      activities: { $all:
... ["Seminar", "Hackathon"] }
...    },
...    {
...      _id: 0,
...      name: 1,
...      activities: 1
...    }
... );  // kanchan kapri // Ragistration no - 1240258215

MyProjectMongodb> |
```

6. Subdocuments and Nested Conditions

Q12. Find students who scored more than 80 in "Web Development" only if they belong to the "Computer Science" department.

```
db.enrollments.find(
... {
...     course_title: "Web Development",
...     marks: { $gt: 80 },
...     department: "Computer Science"
...   },
...   {
...     _id: 0,
...     student_id: 1,
...     marks: 1,
...     course_title: 1,
...     department: 1
...   }
... );
```

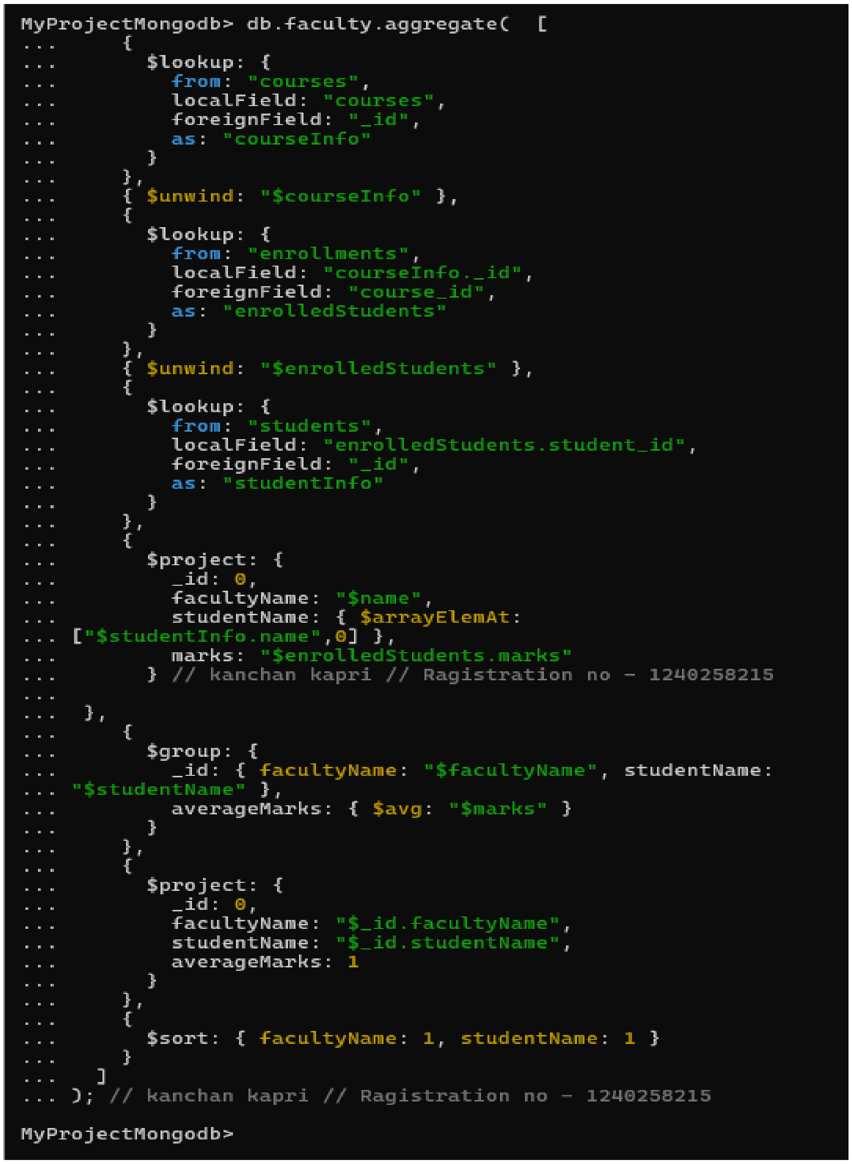


```
MyProjectMongodb> db.enrollments.find(
... {
...     course_title: "Web Development",
...     marks: { $gt: 80 },
...     department: "Computer Science"
...   },
...   {
...     _id: 0,
...     student_id: 1,
...     marks: 1,
...     course_title: 1,
...     department: 1
...   }
... );// kanchan kapri // Ragistration no - 1240258215

MyProjectMongodb> |
```

7. Advanced Aggregation (Challenge Level)

Q13. For each faculty member, list the names of all students enrolled in their courses along with average marks per student per faculty.

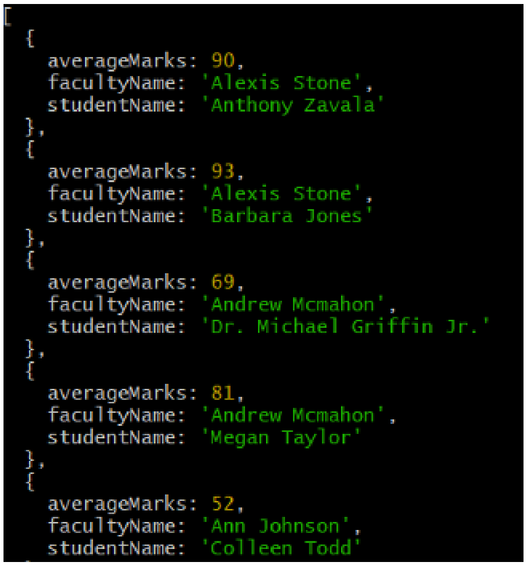
```
db.faculty.aggregate(  [
...    {
...      $lookup: {
...        from: "courses",
...        localField: "courses",
...        foreignField: "_id",
...        as: "courseInfo"
...      }
...    },
...    { $unwind: "$courseInfo" },
...    {
...      $lookup: {
...        from: "enrollments",
...        localField: "courseInfo._id",
...        foreignField: "course_id",
...        as: "enrolledStudents"
...      }
...    },
...    { $unwind: "$enrolledStudents" },
...    {
...      $lookup: {
...        from: "students",
...        localField: "enrolledStudents.student_id",
...        foreignField: "_id",
...        as: "studentInfo"

...      }

...    },

...    {

...      $project: {

...        _id: 0,

...        facultyName: "$name",

...        studentName: { $arrayElemAt:

... ["$studentInfo.name",0] },

...        marks: "$enrolledStudents.marks"

...      }

... },

...    {

...      $group: {

...        _id: { facultyName: "$facultyName", studentName:

... "$studentName" },

...        averageMarks: { $avg: "$marks" }

...      }

...    },

...    {

...      $project: {

...        _id: 0,

...        facultyName: "$_id.facultyName",

...        studentName: "$_id.studentName",

...        averageMarks: 1

...      }

...    },

...    {

...      $sort: { facultyName: 1, studentName: 1 }

...    }

...  ]
```



```
MyProjectMongodb> db.faculty.aggregate(  [
...       {
...         $lookup: {
...           from: "courses",
...           localField: "courses",
...           foreignField: "_id",
...           as: "courseInfo"
...         }
...       },
...       { $unwind: "$courseInfo" },
...       {
...         $lookup: {
...           from: "enrollments",
...           localField: "courseInfo._id",
...           foreignField: "course_id",
...           as: "enrolledStudents"
...         }
...       },
...       { $unwind: "$enrolledStudents" },
...       {
...         $lookup: {
...           from: "students",
...           localField: "enrolledStudents.student_id",
...           foreignField: "_id",
...           as: "studentInfo"
...         }
...       },
...       {
...         $project: {
...           _id: 0,
...           facultyName: "$name",
...           studentName: { $arrayElemAt:
... ["$studentInfo.name",0] },
...           marks: "$enrolledStudents.marks"
...         } // kanchan kapri // Ragistration no - 1240258215
...
...    },
...       {
...         $group: {
...           _id: { facultyName: "$facultyName", studentName:
... "$studentName" },
...           averageMarks: { $avg: "$marks" }
...         }
...       },
...       {
...         $project: {
...           _id: 0,
...           facultyName: "$_id.facultyName",
...           studentName: "$_id.studentName",
...           averageMarks: 1
...         }
...       },
...       {
...         $sort: { facultyName: 1, studentName: 1 }
...       }
...     ]
... ); // kanchan kapri // Ragistration no - 1240258215

MyProjectMongodb>
```

...);

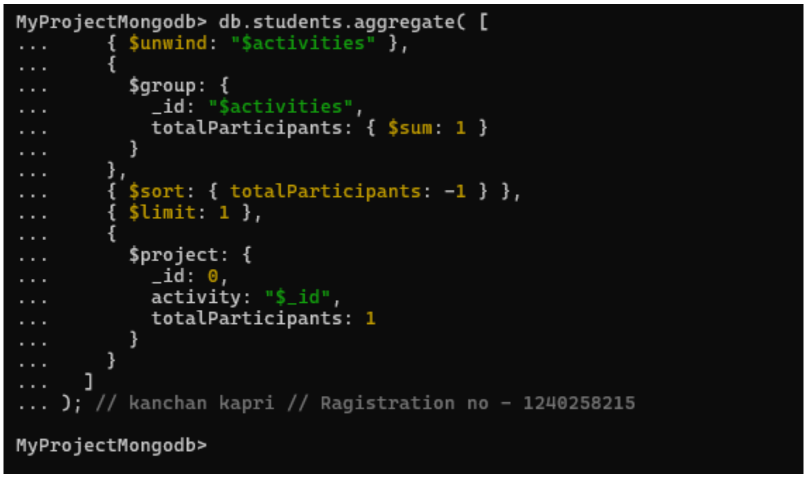
Output :



```
[
  {
    averageMarks: 90,
    facultyName: 'Alexis Stone',
    studentName: 'Anthony Zavala'
  },
  {
    averageMarks: 93,
    facultyName: 'Alexis Stone',
    studentName: 'Barbara Jones'
  },
  {
    averageMarks: 69,
    facultyName: 'Andrew Mcmahon',
    studentName: 'Dr. Michael Griffin Jr.'
  },
  {
    averageMarks: 81,
    facultyName: 'Andrew Mcmahon',
    studentName: 'Megan Taylor'
  },
  {
    averageMarks: 52,
    facultyName: 'Ann Johnson',
    studentName: 'Colleen Todd'
```

Q14. Show the most popular activity type (e.g., Hackathon, Seminar, etc.) by number of student participants.

```
db.students.aggregate( [
...     { $unwind: "$activities" },
...     {
...       $group: {
...         _id: "$activities",
...         totalParticipants: { $sum: 1 }
...       }
...     },
...     { $sort: { totalParticipants: -1 } },
...     { $limit: 1 },
...     {
...       $project: {
...         _id: 0,
...         activity: "$_id",
...         totalParticipants: 1
...       }
...     }
...   ]
... );
```



```
MyProjectMongodb> db.students.aggregate( [
...     { $unwind: "$activities" },
...     {
...       $group: {
...         _id: "$activities",
...         totalParticipants: { $sum: 1 }
...       }
...     },
...     { $sort: { totalParticipants: -1 } },
...     { $limit: 1 },
...     {
...       $project: {
...         _id: 0,
...         activity: "$_id",
...         totalParticipants: 1
...       }
...     }
...   ]
... ); // kanchan kapri // Ragistration no - 1240258215

MyProjectMongodb>
```

--End of Project--