BABU BANARASI DAS UNIVERSITY

ASSIGNMENT-1

NO SQL AND DBAAS(PROJECT)

NAME – MANGESH YADAV

BATCH – BCADS24

CLASS ROLL NO. – 27

UNIVERSITY ROLL NO.- 1240258248

SUBMITED TO: MR. ANKIT VERMA

PROJECT

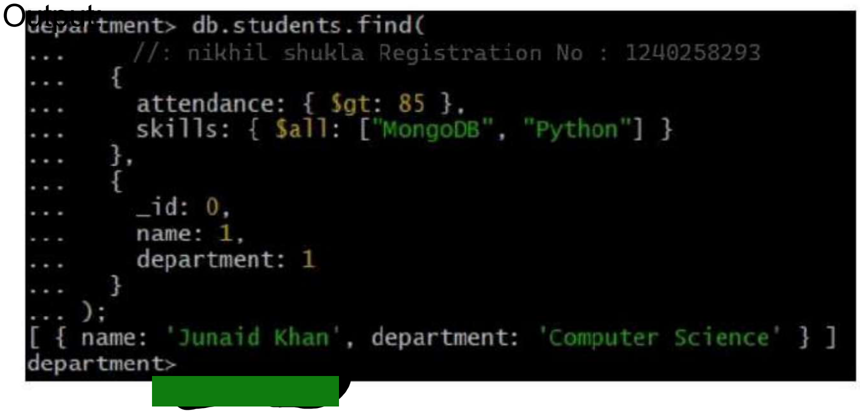
1. Complex Filters & Projections:

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

Solution:

```
   {
        db.students.find(

            attendance: ( $gt: 85 },
         }, skills: ( $al!: ["MongoDB",
        "Python"] }
         {
            _id:

                  0,
         }  name: 1,
            department: 1
        );
```

Output:



```
department> db.students.find(
...     //: nikhil shukla Registration No : 1240258293
...    {
...      attendance: { $gt: 85 },
...      skills: { $all: ["MongoDB", "Python"] }
...    },
...    {
...      _id: 0,
...      name: 1,
...      department: 1
...    }
... );
[ { name: 'Junaid Khan', department: 'Computer Science' } ]
department>
```

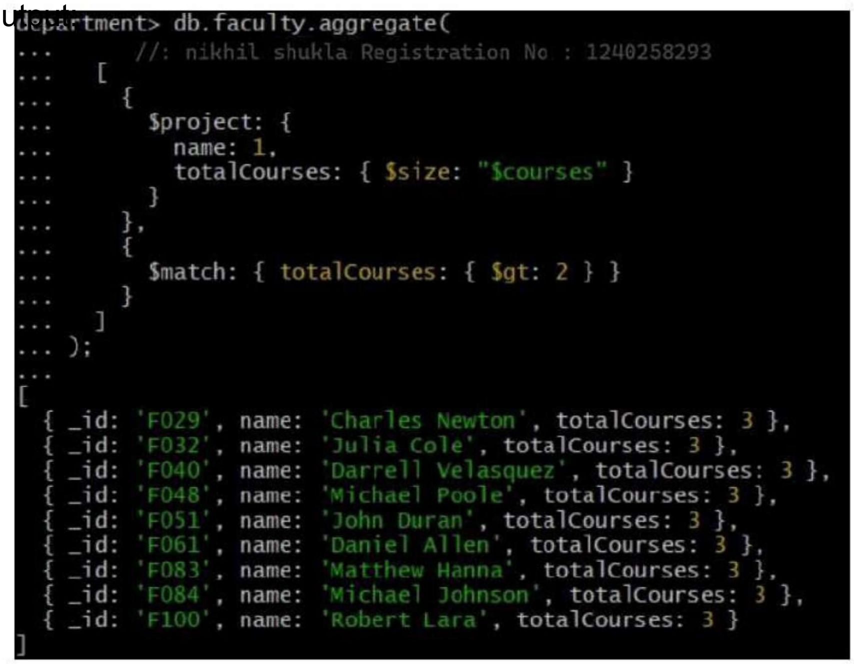
[redacted]

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

Solution:

```
        db.faculty.aggregate(

              $project: (
               name: 1,
                totalCourses: ( $size:
        "$courses


              $match: (
        totalCourses: ( $gt: 2 ) )
          ?]])
```

Output:



```
department> db.faculty.aggregate(
...     //: nikhil shukla Registration No : 1240258293
...    [
...      {
...        $project: {
...          name: 1,
...          totalCourses: { $size: "$courses" }
...        }
...      },
...      {
...        $match: { totalCourses: { $gt: 2 } }
...      }
...    ]
... );
...
[
  { _id: 'F029', name: 'Charles Newton', totalCourses: 3 },
  { _id: 'F032', name: 'Julia Cole', totalCourses: 3 },
  { _id: 'F040', name: 'Darrell Velasquez', totalCourses: 3 },
  { _id: 'F048', name: 'Michael Poole', totalCourses: 3 },
  { _id: 'F051', name: 'John Duran', totalCourses: 3 },
  { _id: 'F061', name: 'Daniel Allen', totalCourses: 3 },
  { _id: 'F083', name: 'Matthew Hanna', totalCourses: 3 },
  { _id: 'F084', name: 'Michael Johnson', totalCourses: 3 },
  { _id: 'F100', name: 'Robert Lara', totalCourses: 3 }
]
```

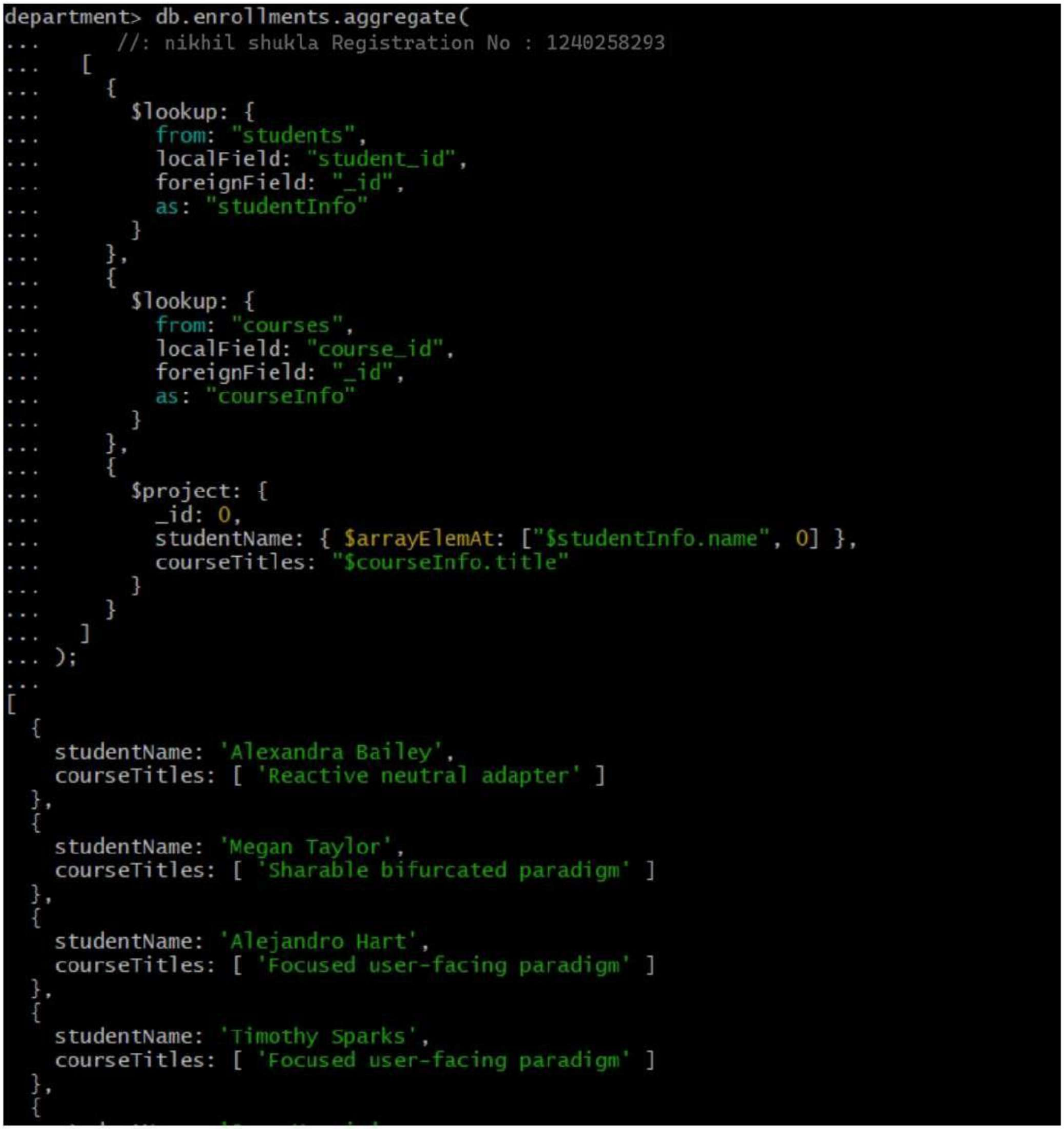
2. Joins ($lookup) and Aggregations:

Q3. Write a query to show each student's name along with the course titles they are enrolled in (use $lookup between enrollments, students, and courses).

Solution:

```
db.enrollments.aggregate(  [


    $lookup: (
      from: "students",
      localF ield: "student_id",
      foreignField: "_id",
      as: "studentInfo"




    $lookup: (
      from: "courses",
      localField: "course_id",
      foreignF ield: "_id",
      as: "courseInfo"




    $project: (
      _id: 0,
      studentName: ( $arrayElemAt: ["$studentInfo.name", 0] },
      courseTides:  "$courseInfo.title"




) I
```

Output:



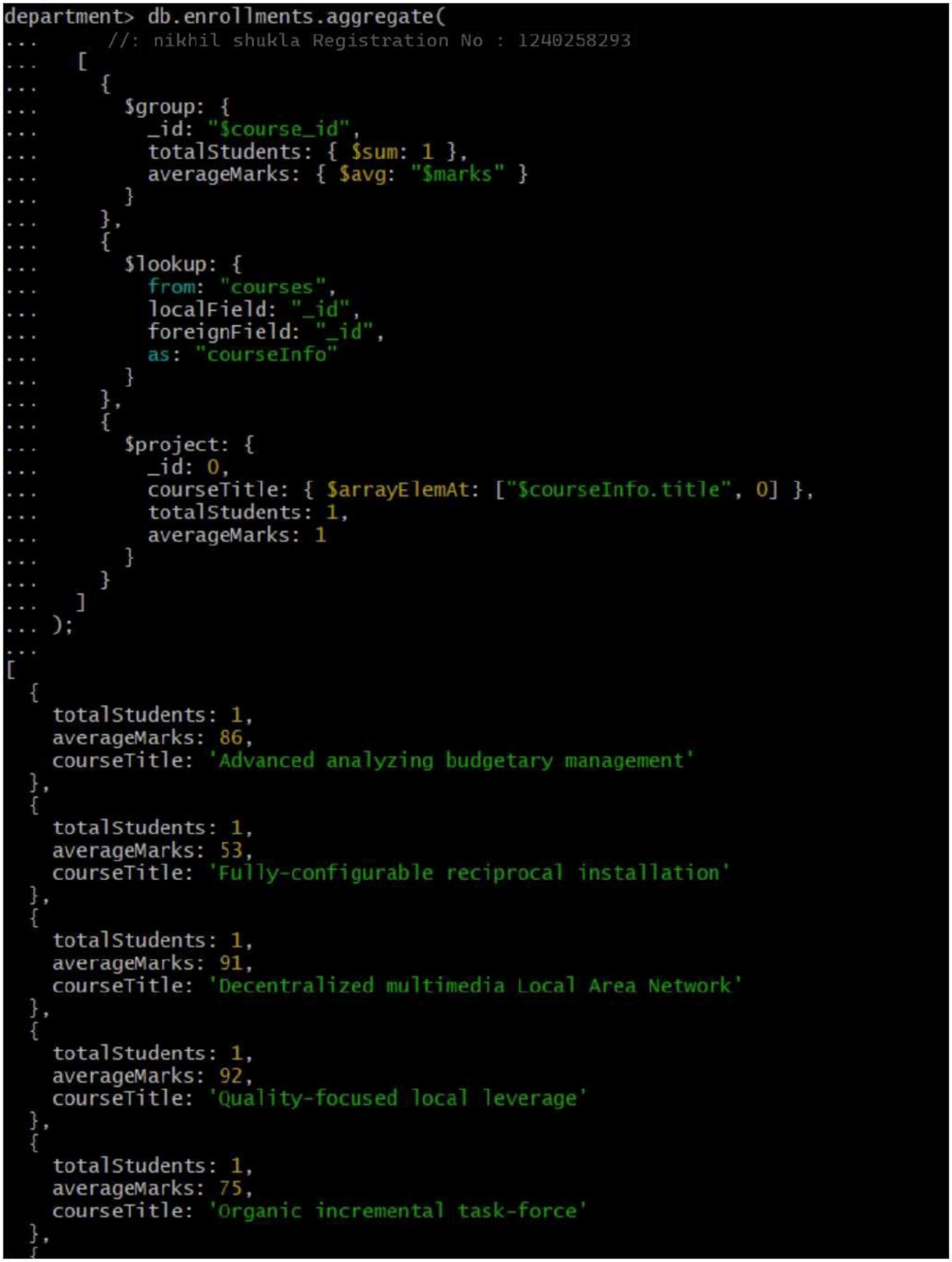
```
department> db.enrollments.aggregate(
...       //: nikhil shukla Registration No : 1240258293
...    [
...      {
...        $lookup: {
...          from: "students",
...          localField: "student_id",
...          foreignField: "_id",
...          as: "studentInfo"
...        }
...      },
...      {
...        $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: [ 'Reactive neutral adapter' ]
  },
  {
    studentName: 'Megan Taylor',
    courseTitles: [ 'Sharable bifurcated paradigm' ]
  },
  {
    studentName: 'Alejandro Hart',
    courseTitles: [ 'Focused user-facing paradigm' ]
  },
  {
    studentName: 'Timothy Sparks',
    courseTitles: [ 'Focused user-facing paradigm' ]
  },
  {
```

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

Solution:

```
     db.enrollments.aggregate(  [


         $group: (
          _id: "$course_id",
          totalstudents: { $sum:  1 },
          averageMarks: ( $avg: "$marks" )




         $lookup: (
          from: "courses",
          localF ield: "_id",
          foreignF ield: "_id",
          as: "courseInfo"




         $project: (
          _id: 0,
          courseTitle: ( $arrayElemAt: ["$courseInfo.title", 0] },
          totalstudents:  1,
          averageMarks:  1




     ) I
```

Output:



```
department> db.enrollments.aggregate(
...     //: nikhil shukla Registration No : 1240258293
...   [
...     {
...       $group: {
...         _id: "$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
...       }
...     }
...   ]
... );
...
[
  {
    totalStudents: 1,
    averageMarks: 86,
    courseTitle: 'Advanced analyzing budgetary management'
  },
  {
    totalStudents: 1,
    averageMarks: 53,
    courseTitle: 'Fully-configurable reciprocal installation'
  },
  {
    totalStudents: 1,
    averageMarks: 91,
    courseTitle: 'Decentralized multimedia Local Area Network'
  },
  {
    totalStudents: 1,
    averageMarks: 92,
    courseTitle: 'Quality-focused local leverage'
  },
  {
    totalStudents: 1,
    averageMarks: 75,
    courseTitle: 'Organic incremental task-force'
  },
  {
```

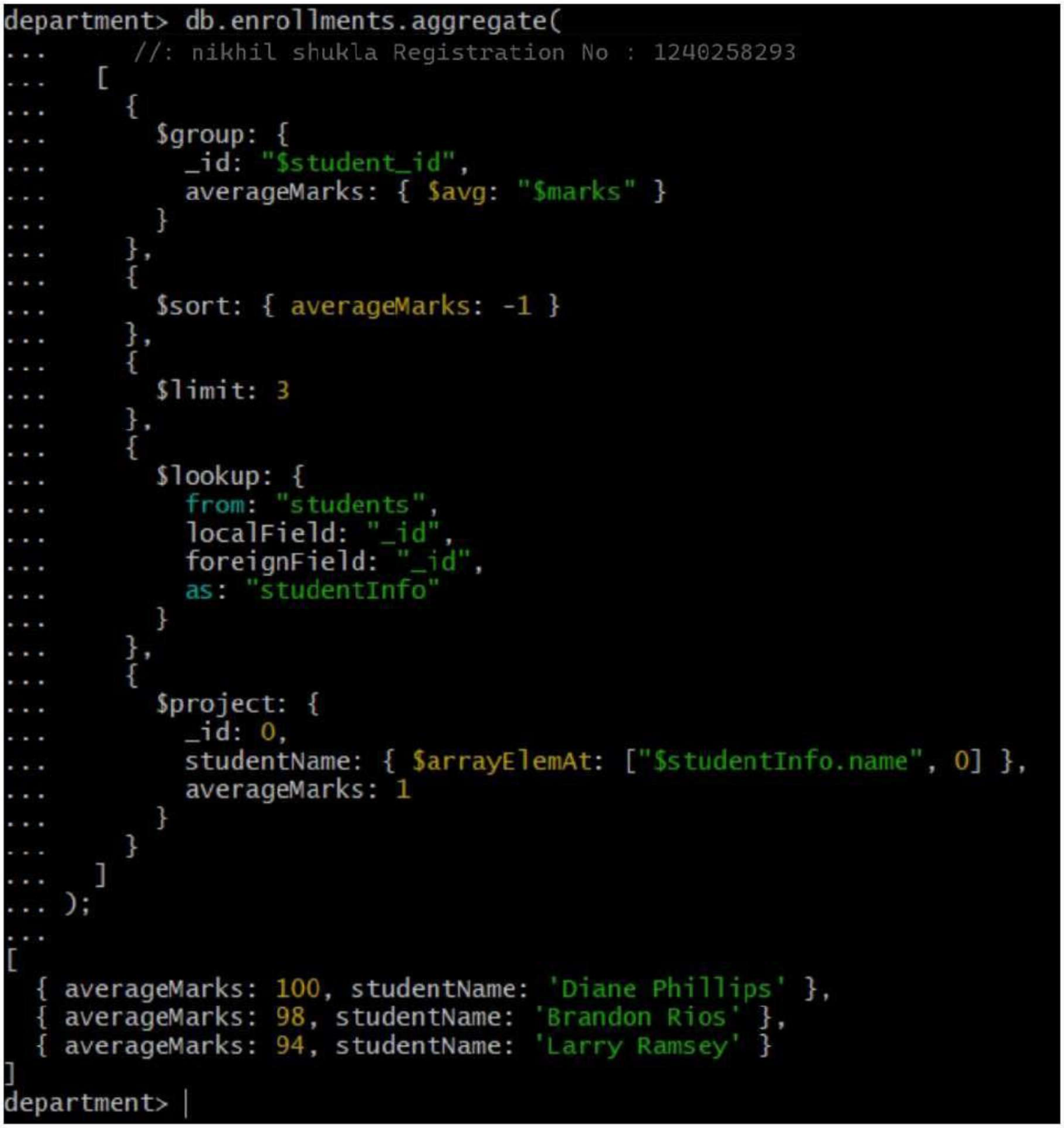
3. Grouping, Sorting, and Limiting:

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

Solution:

```
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", 0j },
      averageMarks: 1




) fi
```

Output:



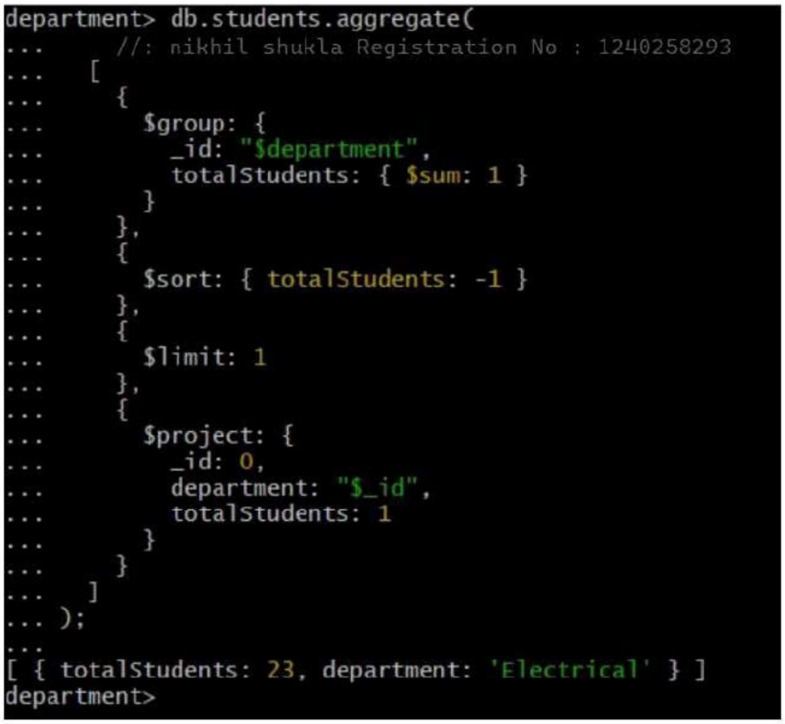
```
department> db.enrollments.aggregate(
...      //: nikhil shukla Registration No : 1240258293
...   [
...     {
...       $group: {
...         _id: "$student_id",
...         averageMarks: { $avg: "$marks" }
...       }
...     },
...     {
...       $sort: { averageMarks: -1 }
...     },
...     {
...       $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: 'Larry Ramsey' }
]
department> |
```

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

Solution:

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

Output:



```
department> db.students.aggregate(
...      //: nikhil shukla Registration No : 1240258293
...    [
...      {
...        $group: {
...          _id: "$department",
...          totalStudents: { $sum: 1 }
...        }
...      },
...      {
...        $sort: { totalStudents: -1 }
...      },
...      {
...        $limit: 1
...      },
...      {
...        $project: {
...          _id: 0,
...          department: "$_id",
...          totalStudents: 1
...        }
...      }
...    ]
... );
...
[ { totalStudents: 23, department: 'Electrical' } ]
department>
```

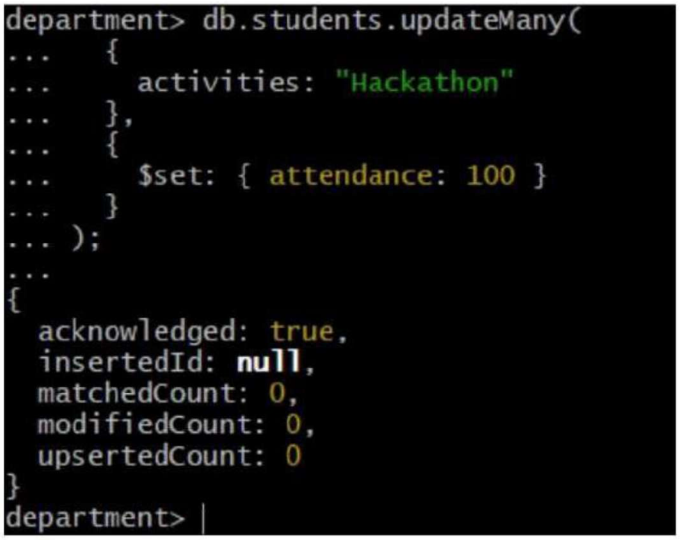
4. Update, Insert, and Delete:

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

Solution:

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

Output:



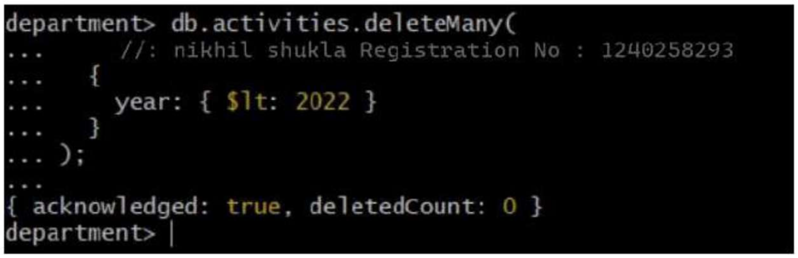
```
department> db.students.updateMany(
...    {
...      activities: "Hackathon"
...    },
...    {
...      $set: { attendance: 100 }
...    }
... );
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
department> |
```

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

Solution:

```
db.activities.deleteMany{
 {
   year: ( SIt: 2022 }
 }
);
```

Output:



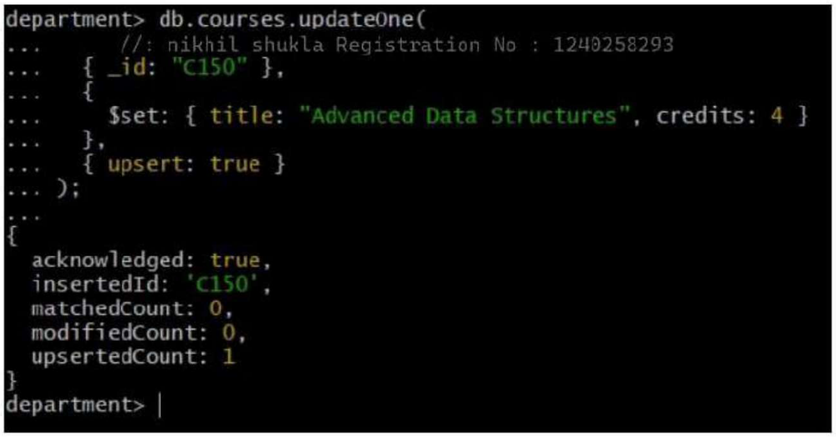
```
department> db.activities.deleteMany(
...      //: nikhil shukla Registration No : 1240258293
...    {
...      year: { $lt: 2022 }
...    }
... );
...
{ acknowledged: true, deletedCount: 0 }
department> |
```

Q9. Insert 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".

Solution:

```
db.courses.updateone(
 { _id: "C150" },
 {
   $set: ( title: "Advanced
Data Structures", credits:
4 )
 },
 ( upsert: true }
);
```

Output:



```
department> db.courses.updateOne(
...      //: nikhil shukla Registration No : 1240258293
...    { _id: "C150" },
...    {
...      $set: { title: "Advanced Data Structures", credits: 4 }
...    },
...    { upsert: true }
... );
...
{
  acknowledged: true,
  insertedId: 'C150',
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
department> |
```

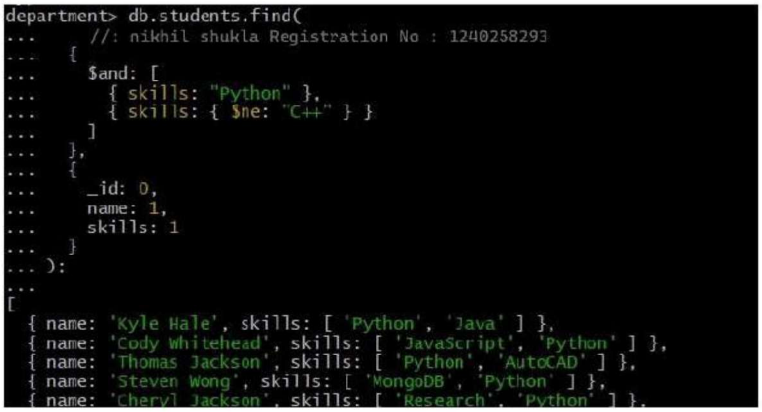
5. Array & Operator Usage:

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

Solution:

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

Output:



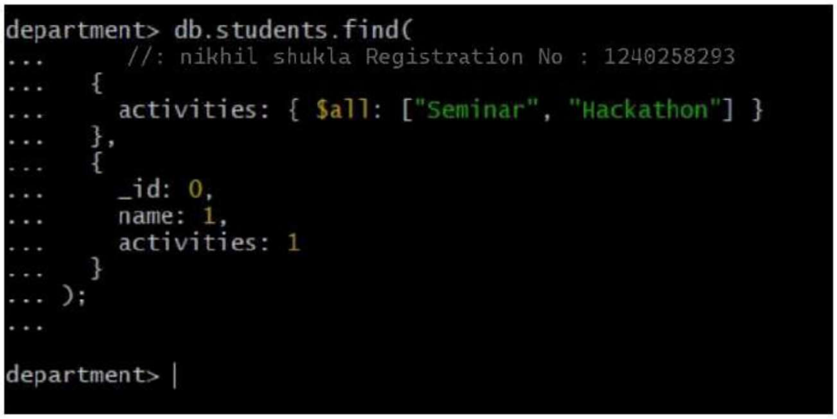
```
department> db.students.find(
...      //: nikhil shukla Registration No : 1240258293
...    {
...      $and: [
...        { skills: "Python" },
...        { skills: { $ne: "C++" } }
...      ]
...    },
...    {
...      _id: 0,
...      name: 1,
...      skills: 1
...    }
... );
...
[
  { name: 'Kyle Hale', skills: [ 'Python', 'Java' ] },
  { name: 'Cody Whitehead', skills: [ 'JavaScript', 'Python' ] },
  { name: 'Thomas Jackson', skills: [ 'Python', 'AutoCAD' ] },
  { name: 'Steven Wong', skills: [ 'MongoDB', 'Python' ] },
  { name: 'Cheryl Jackson', skills: [ 'Research', 'Python' ] },
```

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

Solution:

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

Output:



```
department> db.students.find(
...      //: nikhil shukla Registration No : 1240258293
...   {
...     activities: { $all: ["Seminar", "Hackathon"] }
...   },
...   {
...     _id: 0,
...     name: 1,
...     activities: 1
...   }
... );
...

department> |
```

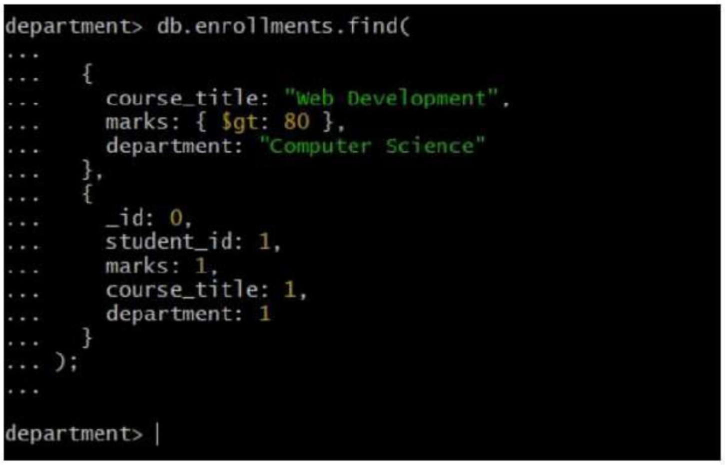
6. Subdocuments and Nested Conditions:

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

Solution:

```
db.enrollments.find(
{
   course_title: "Web Development",
   marks: ( 8gt: 80 },
   departmen t: "Computer Science "
 },
 {
   _id: 0,
   student  id: 1,
   marks: 1,
   course_title: 1,
   department: 1
 }
);
```

Output:



```
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
...    }
... );
...

department> |
```

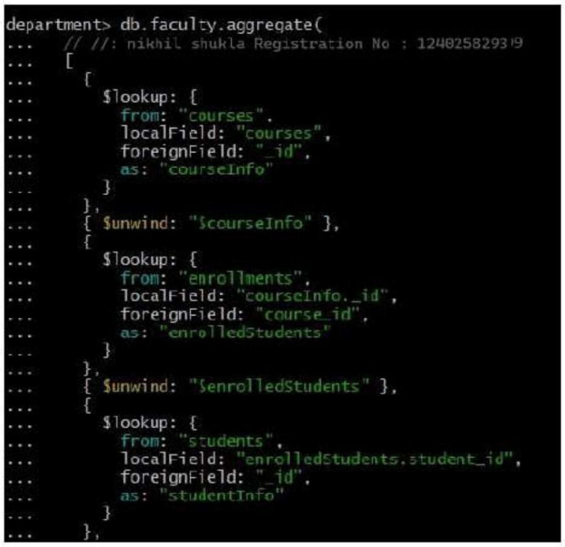
7. Advanced Aggregation (Challenge Level):

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

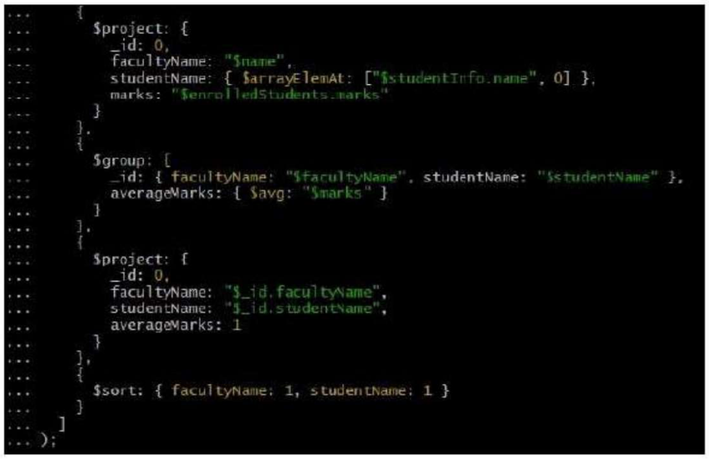
Solution:

```
db.faculty.aggregate(  [
  {
    $lookup: {
      from: "cou rses ",
      localField: "courses",
      foreignField: "_id",
      as: "courseInfo™
    }
  },
  ( $unwind: "$courseInfo" ),
  {
    $lookup {
      from: "enrollments",
      localField: "courseInfo._id",
      foreignField: "course id",
      as: "enrolledstudents"
    }
  },
  ( $unwind: "$enrolledStu dents" ),
  {
    $lookup: (
      from: "students",
      localField: "enrolledStudents.student id",
      foreignField: "_id",
      as: "studentInfo"
    }
  },
  {
    $project: {
      _id: 0,
      facultyName: "Sname",
      studentName: ( $arrayElemAt:
["$studentInfo.name",0] ),
      marks: "$enrolledStudents.marks"
    }
  },
  {
    $group: {
      id: ( facultyName: "$facultyName", studentName:
"$studentName" },
      averageMarks: ( $avg: "$marks" }
    }
  },
  {
    $project: {
      _id: 0,
      facultyName: "$ id.faculty Name",
      studentName: "$  id.studentName",
      averageMarks: 1
    }
  },
  {
    $sort: ( facultyName: 1, studentName: 1 )
  }
 ]
);
```

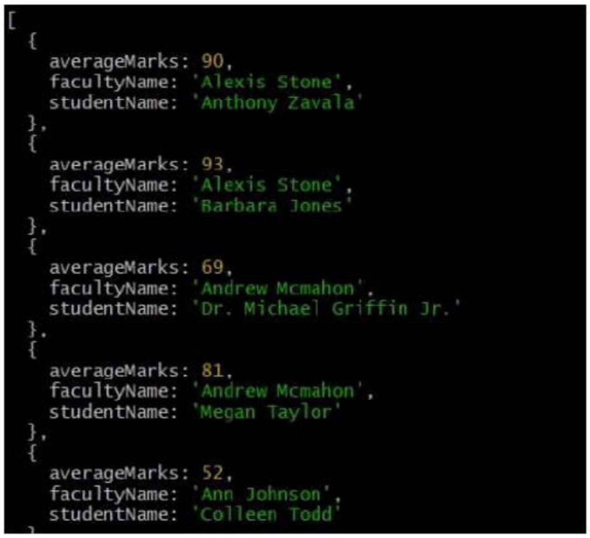
Output:



```
department> db.faculty.aggregate(
...    // //: nikhil shukla Registration No : 12402582939
...    [
...      {
...        $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 }
...      }
...    ]
... );
```



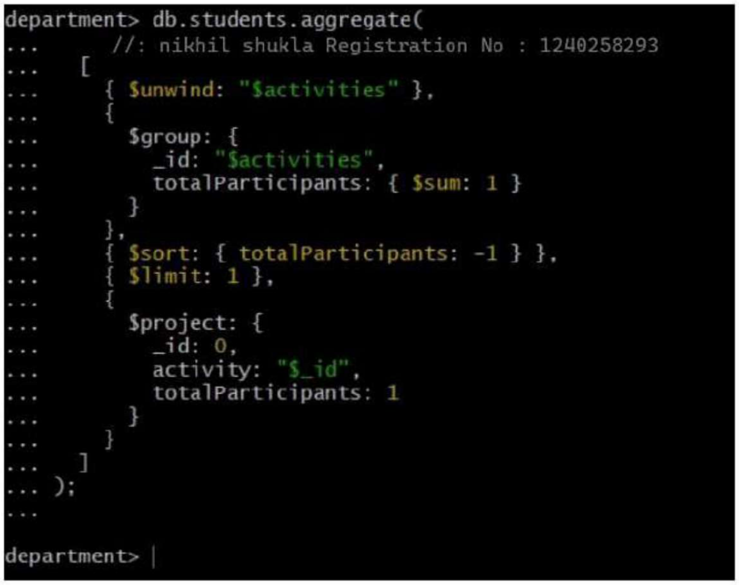
```
[
  {
    averageMarks: 90,
    facultyName: 'Alexis Stone',
    studentName: 'Anthony Zavala'
  },
  {
    averageMarks: 93,
    facultyName: 'Alexis Stone',
    studentName: 'Barbara Jones'
  },
  {
    averageMarks: 69,
    facultyName: 'Andrew Mcmahon',
    studentName: 'Dr. Michael Griffin Jr.'
  },
  {
    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.

Solution:

```
db.students.aggregate( [
   ( $unwind: "8activities" },
   {
     $group: {
       _id: "$activities",
       totalParticipants: ( $sum: 1 )
     }
   },
   ( $sort: ( totalParticipants: -1 } },
   ( $limit: 1 },
   {
     $project: (
       _id: 0,
       activity: "$_id",
       totalParticipants: 1
     }
   }
 ]
);
```

Output:



```
department> db.students.aggregate(
...      //: nikhil shukla Registration No : 1240258293
...    [
...      { $unwind: "$activities" },
...      {
...        $group: {
...          _id: "$activities",
...          totalParticipants: { $sum: 1 }
...        }
...      },
...      { $sort: { totalParticipants: -1 } },
...      { $limit: 1 },
...      {
...        $project: {
...          _id: 0,
...          activity: "$_id",
...          totalParticipants: 1
...        }
...      }
...    ]
... );
...

department> |
```