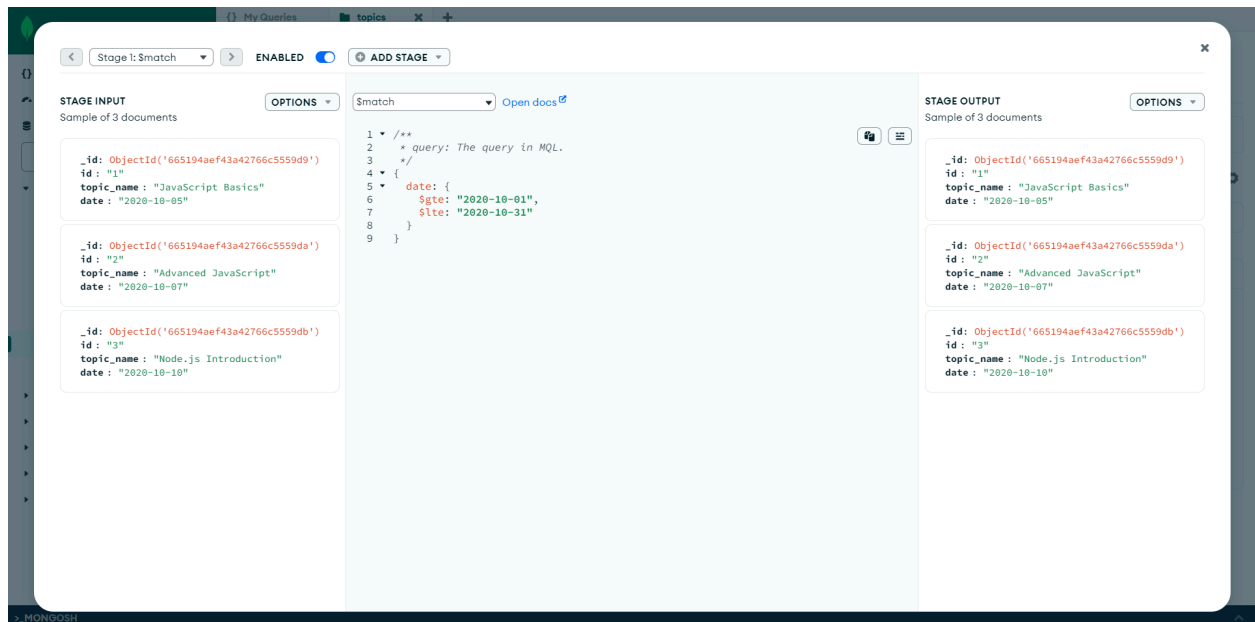


MongoDB Task - 2

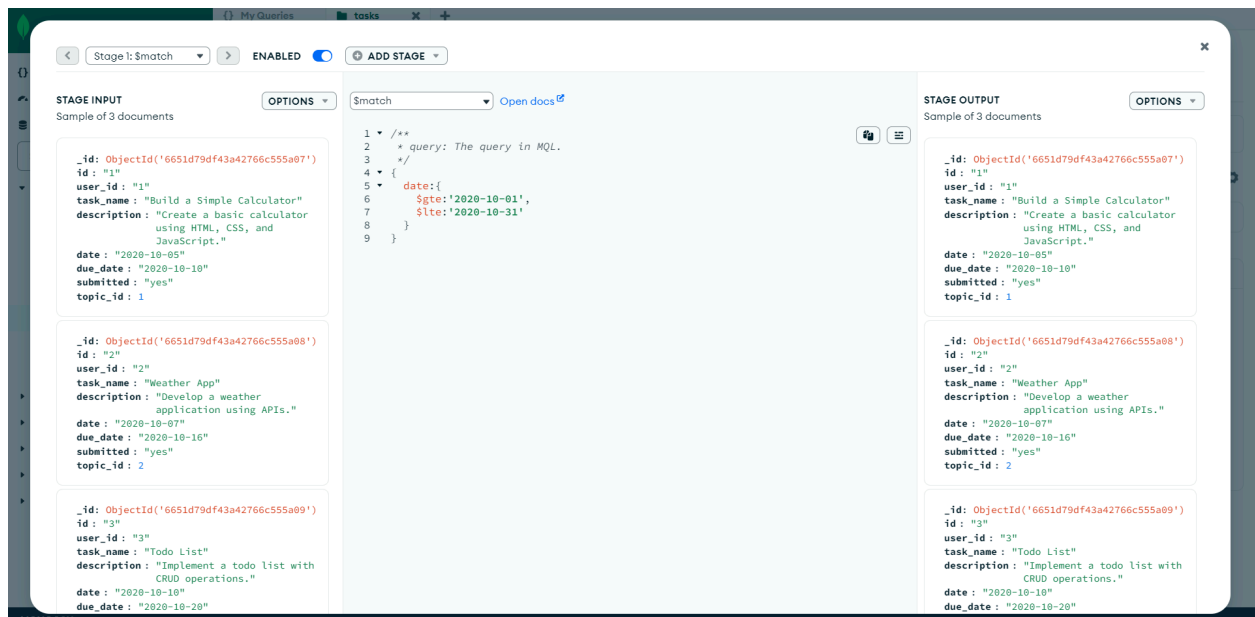
1. Find all the topics and tasks which are taught in the month of October

MongoDB Compass:

Topics were taught in the month of October:



Tasks were given in the month of October



Shell:

Topics were taught in the month of October:

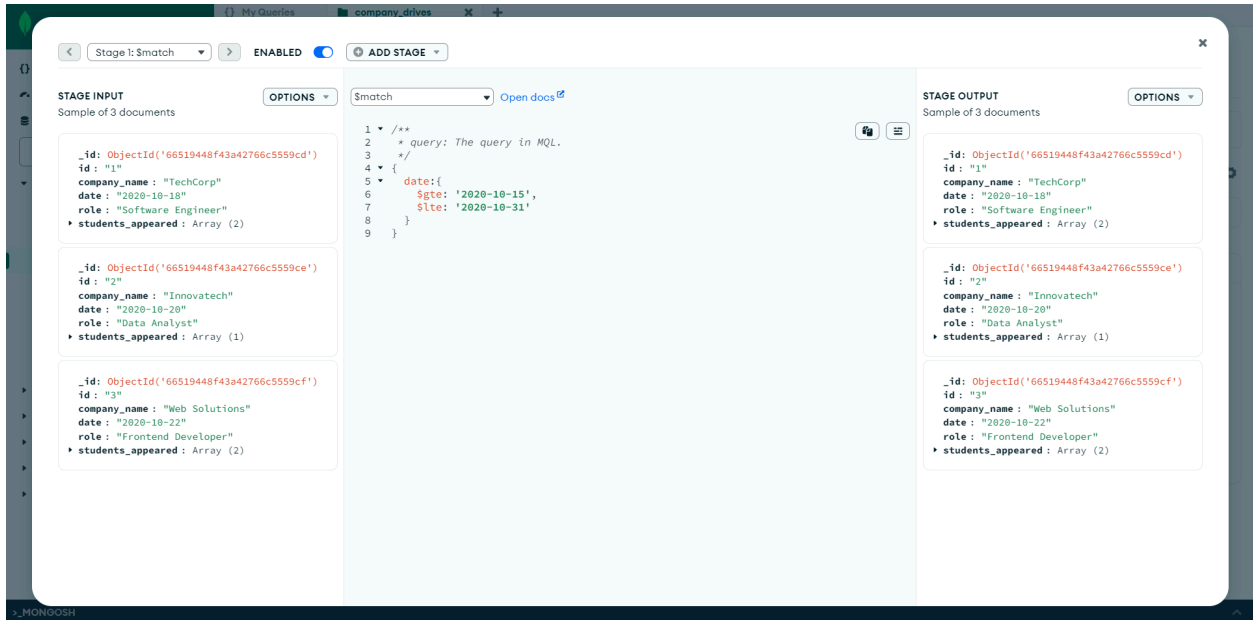
```
Guvi> db.topics.aggregate({$match:{date:{$gte:'2020-10-01', $lte:'2020-10-31'}}})
[
  {
    _id: ObjectId('665194aef43a42766c5559d9'),
    id: '1',
    topic_name: 'JavaScript Basics',
    date: '2020-10-05'
  },
  {
    _id: ObjectId('665194aef43a42766c5559da'),
    id: '2',
    topic_name: 'Advanced JavaScript',
    date: '2020-10-07'
  },
  {
    _id: ObjectId('665194aef43a42766c5559db'),
    id: '3',
    topic_name: 'Node.js Introduction',
    date: '2020-10-10'
  }
]
Guvi> |
```

Tasks were given in the month of October

```
Guvi> db.tasks.aggregate({$match:{date:{$gte:'2020-10-01', $lte:'2020-10-31'}}})
[
  {
    _id: ObjectId('6651d79df43a42766c555a07'),
    id: '1',
    user_id: '1',
    task_name: 'Build a Simple Calculator',
    description: 'Create a basic calculator using HTML, CSS, and JavaScript.',
    date: '2020-10-05',
    due_date: '2020-10-10',
    submitted: 'yes',
    topic_id: 1
  },
  {
    _id: ObjectId('6651d79df43a42766c555a08'),
    id: '2',
    user_id: '2',
    task_name: 'Weather App',
    description: 'Develop a weather application using APIs.',
    date: '2020-10-07',
    due_date: '2020-10-16',
    submitted: 'yes',
    topic_id: 2
  },
  {
    _id: ObjectId('6651d79df43a42766c555a09'),
    id: '3',
    user_id: '3',
    task_name: 'Todo List',
    description: 'Implement a todo list with CRUD operations.',
    date: '2020-10-10',
    due_date: '2020-10-20',
    submitted: 'no',
    topic_id: 3
  }
]
Guvi> |
```

2. Find all the company drives which appeared between 15 oct-2020 and 31-oct-2020

MongoDB Compass:



Shell:

```
Guvi> db.company_drives.aggregate({$match:{date:{$gte:'2020-10-15', $lte:'2020-10-31'}}})
[
  {
    _id: ObjectId('66519448f43a42766c5559cd'),
    id: '1',
    company_name: 'TechCorp',
    date: '2020-10-18',
    role: 'Software Engineer',
    students_appeared: [ 1, 2 ]
  },
  {
    _id: ObjectId('66519448f43a42766c5559ce'),
    id: '2',
    company_name: 'Innovatech',
    date: '2020-10-20',
    role: 'Data Analyst',
    students_appeared: [ 1 ]
  },
  {
    _id: ObjectId('66519448f43a42766c5559cf'),
    id: '3',
    company_name: 'Web Solutions',
    date: '2020-10-22',
    role: 'Frontend Developer',
    students_appeared: [ 1, 3 ]
  }
]
Guvi>
```

3. Find all the company drives and students who are appeared for the placement

MongoDB Compass:

Lookup stage:

The screenshot shows the MongoDB Compass interface with a query pipeline. The first stage is a Lookup stage, which is enabled. The stage input is a sample of 3 documents. The stage output is a sample of 3 documents. The stage configuration is as follows:

```
1 /**
2  * from: The target collection.
3  * localField: The local join field.
4  * foreignField: The target join field.
5  * as: The name for the results.
6  * pipeline: Optional pipeline to run on the foreign collection.
7  * let: Optional variables to use in the pipeline field stages.
8  */
9 {
10   from: 'users',
11   localField: 'students_appeared',
12   foreignField: 'id',
13   as: 'students'
14 }
```

The stage output shows the result of the lookup operation. It contains 3 documents, each with an `students_appeared` array and a `students` array. The `students` array contains the details of the students who appeared for the placement.

Unwind:

The screenshot shows the MongoDB Compass interface with a query pipeline. The second stage is an Unwind stage, which is enabled. The stage input is a sample of 3 documents. The stage output is a sample of 5 documents. The stage configuration is as follows:

```
1 /**
2  * path: Path to the array field.
3  * includeArrayIndex: Optional name for index.
4  * preserveNullAndEmptyArrays: Optional
5  * toggle to unwind null and empty values.
6  */
7 {
8   path: '$students'
9 }
```

The stage output shows the result of the unwind operation. It contains 5 documents, each with an `students_appeared` array and a `students` object. The `students` object contains the details of the student who appeared for the placement.

Project:

The screenshot shows the MongoDB Atlas Query Editor interface. The top navigation bar includes a back arrow, a dropdown menu set to 'Stage 3: \$project', a toggle for 'ENABLED', and an 'ADD STAGE' button. The main workspace is divided into three panels:

- STAGE INPUT:** Labeled 'Sample of 5 documents', it displays four document snippets. Each document has fields like `_id`, `company_name`, `date`, `role`, `students_appeared`, and `students`.
- Query:** The central panel shows a JavaScript-style query using the `$project` operator. It includes comments: `1 /**`, `2 * specifications: The fields to`, `3 * include or exclude.`, `4 */`, and a JSON object: `5 {`, `6 _id: 0,`, `7 company_name: 1,`, `8 role: 1,`, `9 'students.name': 1,`, `10 'students.email': 1`, `11 }`.
- STAGE OUTPUT:** Labeled 'Sample of 5 documents', it shows the resulting documents after the `$project` stage. The documents are filtered and formatted according to the query, showing only the specified fields.

Shell:

```
Guvi> db.company_drives.aggregate([{$lookup:{from:"users", localField:"students_appeared", foreignField:"id", as:"students"}}])
[
  {
    _id: ObjectId('66519448f43a42766c5559cd'),
    id: '1',
    company_name: 'TechCorp',
    date: '2020-10-18',
    role: 'Software Engineer',
    students_appeared: [ 1, 2 ],
    students: [
      {
        _id: ObjectId('665194cff43a42766c5559dd'),
        id: 1,
        name: 'Alice',
        email: 'alice@example.com',
        mentor_id: 1
      },
      {
        _id: ObjectId('665194cff43a42766c5559de'),
        id: 2,
        name: 'Bob',
        email: 'bob@example.com',
        mentor_id: 1
      }
    ]
  },
  {
    _id: ObjectId('66519448f43a42766c5559ce'),
    id: '2',
    company_name: 'Innovatech',
    date: '2020-10-20',
    role: 'Data Analyst',
    students_appeared: [ 1 ],
    students: [
      {
        _id: ObjectId('665194cff43a42766c5559dd'),
        id: 1,
        name: 'Alice',
        email: 'alice@example.com',
        mentor_id: 1
      }
    ]
  }
],
```

```

{
  _id: ObjectId('66519448f43a42766c5559ce'),
  id: '2',
  company_name: 'Innovatech',
  date: '2020-10-20',
  role: 'Data Analyst',
  students_appeared: [ 1 ],
  students: [
    {
      _id: ObjectId('665194cff43a42766c5559dd'),
      id: 1,
      name: 'Alice',
      email: 'Alice@example.com',
      mentor_id: 1
    }
  ]
},
{
  _id: ObjectId('66519448f43a42766c5559cf'),
  id: '3',
  company_name: 'Web Solutions',
  date: '2020-10-22',
  role: 'Frontend Developer',
  students_appeared: [ 1, 3 ],
  students: [
    {
      _id: ObjectId('665194cff43a42766c5559dd'),
      id: 1,
      name: 'Alice',
      email: 'Alice@example.com',
      mentor_id: 1
    },
    {
      _id: ObjectId('665194cff43a42766c5559df'),
      id: 3,
      name: 'Martin',
      email: 'Martin@example.com',
      mentor_id: 2
    }
  ]
}
]
}
Guvi>

```

4. Find the number of problems solved by the user in codekata

MongoDB Compass:

The screenshot shows the MongoDB Compass interface with a pipeline stage named 'Stage1: \$group' enabled. The stage input shows 5 documents, and the stage output shows 3 documents. The pipeline script in the center calculates the total problems solved for each user_id.

STAGE INPUT
Sample of 5 documents

```

{ "_id": "ObjectId('6651afe9f43a42766c5559eb')", "id": "1", "user_id": "1", "module": "Array", "problems_solved": 30 }
{ "_id": "ObjectId('6651afe9f43a42766c5559ec')", "id": "2", "user_id": "2", "module": "Array", "problems_solved": 40 }
{ "_id": "ObjectId('6651afe9f43a42766c5559ed')", "id": "3", "user_id": "3", "module": "Array", "problems_solved": 50 }
{ "_id": "ObjectId('6651afe9f43a42766c5559ee')", "id": "4", "user_id": "1", "module": "Loop", "problems_solved": 30 }
{ "_id": "ObjectId('6651afe9f43a42766c5559ef')", "id": "5", "user_id": "2", "module": "Loop", "problems_solved": 30 }

```

STAGE OUTPUT
Sample of 3 documents

```

{ "_id": "1", "total_problem_solved": 60 }
{ "_id": "3", "total_problem_solved": 50 }
{ "_id": "2", "total_problem_solved": 70 }

```

STAGE SCRIPT

```

1 // **
2 * _id: The id of the group.
3 * fieldN: The first field name.
4 */
5 {
6   _id: '$user_id',
7   total_problem_solved: {
8     $sum: '$problems_solved'
9   }
10 }

```

Shell:

```
Guvi> db.codekata.aggregate([{$group:{$_id:$user_id', total_problem_solved:{$sum: "$problems_solved"}}}])
[
  { _id: '2', total_problem_solved: 70 },
  { _id: '3', total_problem_solved: 50 },
  { _id: '1', total_problem_solved: 60 }
]
Guvi> |
```

5.Find all the mentors with who has the mentee's count more than 15

MongoDB Compass:

localhost:27017 > Guvi > mentors

Documents 3 Aggregations Schema Indexes 1 Validation

{ \$expr: { \$gt: [{ \$size: "\$mentees" }, 15] } }

Generate query Explain Reset Find </> Options

ADD DATA EXPORT DATA UPDATE DELETE 1 - 2 of 2

```
{
  "_id": ObjectId('6651c098f43a42766c5559fa'),
  "id": "1",
  "name": "Dr. Smith",
  "expertise": "JavaScript, Node.js",
  "email": "smith@mentors.com",
  "mentees": Array (18)
}
```

```
{
  "_id": ObjectId('6651c098f43a42766c5559fc'),
  "id": "3",
  "name": "Mr. Lee",
  "expertise": "DevOps, AWS",
  "email": "lee@mentors.com",
  "mentees": Array (20)
}
```

Shell:

```
Guvi> db.mentors.find({$expr:{$gt:[$size:"$mentees"],15}})
[
  {
    _id: ObjectId('6651c098f43a42766c5559fa'),
    id: '1',
    name: 'Dr. Smith',
    expertise: 'JavaScript, Node.js',
    email: 'smith@mentors.com',
    mentees: [
      1, 2, 3, 4, 5, 6, 7,
      8, 9, 10, 11, 12, 13, 14,
      15, 16, 17, 18
    ]
  },
  {
    _id: ObjectId('6651c098f43a42766c5559fc'),
    id: '3',
    name: 'Mr. Lee',
    expertise: 'DevOps, AWS',
    email: 'lee@mentors.com',
    mentees: [
      1, 2, 3, 4, 5, 6, 7,
      8, 9, 10, 11, 12, 13, 14,
      15, 16, 17, 18, 19, 20
    ]
  }
]
Guvi> |
```

6. Find the number of users who are absent and task is not submitted between 15 oct-2020 and 31-oct-2020

MongoDB Compass:

1. \$match Stage

The screenshot shows the MongoDB Compass interface with a query pipeline. The 'Stage 1: \$match' stage is selected and enabled. The 'STAGE INPUT' section shows a sample of 3 documents. The 'STAGE OUTPUT' section shows a sample of 1 document.

STAGE INPUT
Sample of 3 documents

```
{ "_id": ObjectId('66519419f43a42766c5559c7'), "id": "1", "user_id": "1", "date": "2020-10-16", "status": "present" }
{ "_id": ObjectId('66519419f43a42766c5559c7'), "id": "2", "user_id": "2", "date": "2020-10-17", "status": "present" }
{ "_id": ObjectId('66519419f43a42766c5559c7'), "id": "3", "user_id": "3", "date": "2020-10-18", "status": "absent" }
```

STAGE OUTPUT
Sample of 1 document

```
{ "_id": ObjectId('66519419f43a42766c5559c7'), "id": "3", "user_id": "3", "date": "2020-10-18", "status": "absent" }
```

2. \$lookup Stage:

The screenshot shows the MongoDB Compass interface with a query pipeline. The 'Stage 2: \$lookup' stage is selected and enabled. The 'STAGE INPUT' section shows a sample of 1 document. The 'STAGE OUTPUT' section shows a sample of 1 document.

STAGE INPUT
Sample of 1 document

```
{ "_id": ObjectId('66519419f43a42766c5559c7'), "id": "3", "user_id": "3", "date": "2020-10-18", "status": "absent" }
```

STAGE OUTPUT
Sample of 1 document

```
{ "_id": ObjectId('66519419f43a42766c5559c7'), "id": "3", "user_id": "3", "date": "2020-10-18", "status": "absent", "tasks": [ { "_id": ObjectId('6651d79df43a42766c5559c7'), "id": "3", "user_id": "3", "task_name": "Todo List", "description": "Implement a todo list with CRUD operations.", "date": "2020-10-10", "due_date": "2020-10-20", "submitted": "no", "topic_id": 3 } ] }
```


3.\$unwind stage

The screenshot shows the MongoDB Compass interface for a query named "attendance". The "Stages" tab is active, showing "Stage 3: \$unwind". The stage is enabled. The "STAGE INPUT" section shows a sample document with a nested array of tasks. The "STAGE OUTPUT" section shows the result after the \$unwind stage, where the array has been flattened into individual documents.

STAGE INPUT
Sample of 1 document

```
{
  "_id": ObjectId("66519419f43a42766c5559c7"),
  "id": "3",
  "user_id": "3",
  "date": "2020-10-18",
  "status": "absent",
  "tasks": Array(1)
}
```

STAGE OUTPUT
Sample of 1 document

```
{
  "_id": ObjectId("66519419f43a42766c5559c7"),
  "id": "3",
  "user_id": "3",
  "date": "2020-10-18",
  "status": "absent",
  "tasks": {
    "_id": ObjectId("6651d79df43a42766c555a09..."),
    "id": "3",
    "user_id": "3",
    "task_name": "Todo List",
    "description": "Implement a todo list with CRUD operations.",
    "date": "2020-10-18",
    "due_date": "2020-10-20",
    "submitted": "no",
    "topic_id": 3
  }
}
```

4.\$project stage:

The screenshot shows the MongoDB Compass interface for a query named "attendance". The "Stages" tab is active, showing "Stage 4: \$project". The stage is enabled. The "STAGE INPUT" section shows a sample document with a nested array of tasks. The "STAGE OUTPUT" section shows the result after the \$project stage, where the document has been projected to include only the specified fields.

STAGE INPUT
Sample of 1 document

```
{
  "_id": ObjectId("66519419f43a42766c5559c7"),
  "id": "3",
  "user_id": "3",
  "date": "2020-10-18",
  "status": "absent",
  "tasks": Object
}
```

STAGE OUTPUT
Sample of 1 document

```
{
  "id": "3",
  "user_id": "3",
  "status": "absent",
  "tasks": Object
}
```

5.\$group stage:

The screenshot displays the MongoDB Atlas query editor interface for a collection named 'attendance'. The query is configured with a single stage, 'Stage 5: \$group', which is currently enabled. The stage is set to group documents by the field '\$group'. The input document shows a task entry for user ID 3, and the output document shows the grouped result with a count of 1.

STAGE INPUT
Sample of 1 document

```
{
  "id": "3",
  "user_id": "3",
  "status": "absent",
  "tasks": {
    "id": "3",
    "user_id": "3",
    "task_name": "Todo List",
    "due_date": "2020-10-20",
    "submitted": "no"
  }
}
```

STAGE OUTPUT
Sample of 1 document

```
{
  "_id": "3",
  "count": 1
}
```

User ID 3 who was absent and who did not submit the task between 15 oct-2020 and 31-oct-2020.