

User Collection:

```
> db.users.insertMany([
  {
    "name": "Thiru",
    "email": "thiru@mail.com",
    "batch": "B17",
    "task_id": ObjectId(),
    "mentor_id": ObjectId(),
    "codekata": ObjectId(),
    "attendance": 80
  },
  {
    "name": "Arun",
    "email": "arun@mail.com",
    "batch": "B18",
    "task_id": ObjectId(),
    "mentor_id": ObjectId(),
    "codekata": ObjectId(),
    "attendance": 100
  },
  {
    "name": "Deepan",
    "email": "deepan@mail.com",
    "batch": "B19",
    "task_id": ObjectId(),
    "mentor_id": ObjectId(),
    "codekata": ObjectId(),
    "attendance": 100
  },
]);
```

```
> db.users.find()
< {
  _id: ObjectId('67cd879635e99230677ed8d6'),
  name: 'Thiru',
  email: 'thiru@mail.com',
  batch: 'B17',
  task_id: ObjectId('67cd879635e99230677ed8cd'),
  mentor_id: ObjectId('67cd879635e99230677ed8ce'),
  codekata: ObjectId('67cd879635e99230677ed8cf'),
  attendance: 80
}
{
  _id: ObjectId('67cd879635e99230677ed8d7'),
  name: 'Arun',
  email: 'arun@mail.com',
  batch: 'B18',
  task_id: ObjectId('67cd879635e99230677ed8d0'),
  mentor_id: ObjectId('67cd879635e99230677ed8d1'),
  codekata: ObjectId('67cd879635e99230677ed8d2'),
  attendance: 100
}
{
  _id: ObjectId('67cd879635e99230677ed8d8'),
  name: 'Deepan',
  email: 'deepan@mail.com',
  batch: 'B19',
  task_id: ObjectId('67cd879635e99230677ed8d3'),
  mentor_id: ObjectId('67cd879635e99230677ed8d4'),
  codekata: ObjectId('67cd879635e99230677ed8d5'),
  attendance: 100
}
```

Codekata Collection:

```
db.codekata.insertMany([
  {
    user_id: ObjectId('67cd879635e99230677ed8d6'),
    solved: 60
  },
  {
    user_id: ObjectId('67cd879635e99230677ed8d7'),
    solved: 62
  },
  {
    user_id: ObjectId('67cd879635e99230677ed8d8'),
    solved: 100
  }
]);
```

```
< {
  _id: ObjectId('67cd9c8835e99230677ed8d9'),
  user_id: ObjectId('67cd879635e99230677ed8d6'),
  solved: 60
}
{
  _id: ObjectId('67cd9c8835e99230677ed8da'),
  user_id: ObjectId('67cd879635e99230677ed8d7'),
  solved: 62
}
{
  _id: ObjectId('67cd9c8835e99230677ed8db'),
  user_id: ObjectId('67cd879635e99230677ed8d8'),
  solved: 100
}
```

## Attendance Collection:

```
> db.attendance.insertMany([
  {
    user_id: ObjectId('67cd879635e99230677ed8d6'),
    date: ISODate('2020-10-11'),
    status: 'Present'
  },
  {
    user_id: ObjectId('67cd879635e99230677ed8d7'),
    date: ISODate('2020-10-11'),
    status: 'Absent'
  },
  {
    user_id: ObjectId('67cd879635e99230677ed8d8'),
    date: ISODate('2020-10-11'),
    status: 'Present'
  }
]);
```

```
< {
  _id: ObjectId('67cd9d4635e99230677ed8dc'),
  user_id: ObjectId('67cd879635e99230677ed8d6'),
  date: 2020-10-11T00:00:00.000Z,
  status: 'Present'
}
{
  _id: ObjectId('67cd9d4635e99230677ed8dd'),
  user_id: ObjectId('67cd879635e99230677ed8d7'),
  date: 2020-10-11T00:00:00.000Z,
  status: 'Absent'
}
{
  _id: ObjectId('67cd9d4635e99230677ed8de'),
  user_id: ObjectId('67cd879635e99230677ed8d8'),
  date: 2020-10-11T00:00:00.000Z,
  status: 'Present'
}
```

## Topics Collections:

```
> db.topics.insertMany([
  {
    topic: "HTML, CSS, JS",
    date: ISODate("2020-10-01")
  },
  {
    topic: "React JS",
    date: ISODate("2020-11-08")
  },
  {
    topic: "Database",
    date: ISODate("2020-12-15")
  },
  {
    topic: "Node JS",
    date: ISODate("2020-12-22")
  }
]);
```

```
< {
  _id: ObjectId('67cd9e1335e99230677ed8df'),
  topic: 'HTML, CSS, JS',
  date: 2020-10-01T00:00:00.000Z
}
{
  _id: ObjectId('67cd9e1335e99230677ed8e0'),
  topic: 'React JS',
  date: 2020-11-08T00:00:00.000Z
}
{
  _id: ObjectId('67cd9e1335e99230677ed8e1'),
  topic: 'Database',
  date: 2020-12-15T00:00:00.000Z
}
{
  _id: ObjectId('67cd9e1335e99230677ed8e2'),
  topic: 'Node JS',
  date: 2020-12-22T00:00:00.000Z
}
```

Tasks Collections:

```
> db.tasks.insertMany([
  {
    task_id: ObjectId(),
    task_name: "HTML, CSS, and JS Task",
    date: ISODate("2020-10-01"),
    status: "Completed"
  },
  {
    task_id: ObjectId(),
    task_name: "Advance JS Task",
    date: ISODate("2020-10-10"),
    status: "Pending"
  },
  {
    task_id: ObjectId(),
    task_name: "React Task",
    date: ISODate("2020-11-10"),
    status: "Pending"
  },
  {
    task_id: ObjectId(),
    task_name: "Database Task",
    date: ISODate("2020-12-17"),
    status: "Completed"
  },
  {
    task_id: ObjectId(),
    task_name: "Node JS Task",
    date: ISODate("2020-12-24"),
    status: "Completed"
  }
]);
```

```
> db.tasks.find()
< {
  _id: ObjectId('67cd9f7835e99230677ed8e8'),
  task_id: ObjectId('67cd9f7735e99230677ed8e3'),
  task_name: 'HTML, CSS, and JS Task',
  date: 2020-10-01T00:00:00.000Z,
  status: 'Completed'
}
{
  _id: ObjectId('67cd9f7835e99230677ed8e9'),
  task_id: ObjectId('67cd9f7735e99230677ed8e4'),
  task_name: 'Advance JS Task',
  date: 2020-10-10T00:00:00.000Z,
  status: 'Pending'
}
{
  _id: ObjectId('67cd9f7835e99230677ed8ea'),
  task_id: ObjectId('67cd9f7735e99230677ed8e5'),
  task_name: 'React Task',
  date: 2020-11-10T00:00:00.000Z,
  status: 'Pending'
}
{
  _id: ObjectId('67cd9f7835e99230677ed8eb'),
  task_id: ObjectId('67cd9f7735e99230677ed8e6'),
  task_name: 'Database Task',
  date: 2020-12-17T00:00:00.000Z,
  status: 'Completed'
}
{
  _id: ObjectId('67cd9f7835e99230677ed8ec'),
  task_id: ObjectId('67cd9f7735e99230677ed8e7'),
  task_name: 'Node JS Task',
  date: 2020-12-24T00:00:00.000Z,
  status: 'Completed'
}
```

## Company Drives Collection

```
> db.company_drives.insertMany([
  {
    comp_name: "Guvi",
    drive_date: ISODate("2020-10-20"),
    user_id: ObjectId("67cd879635e99230677ed8d6")
  },
  {
    comp_name: "Google",
    drive_date: ISODate("2020-11-15"),
    user_id: ObjectId("67cd879635e99230677ed8d7")
  },
  {
    comp_name: "Microsoft",
    drive_date: ISODate("2020-12-10"),
    user_id: ObjectId("67cd879635e99230677ed8d8")
  }
]);
```

```
> db.company_drives.find()
< {
  _id: ObjectId('67cda32535e99230677ed8ed'),
  comp_name: 'Guvi',
  drive_date: 2020-10-20T00:00:00.000Z,
  user_id: ObjectId('67cd879635e99230677ed8d6')
}
{
  _id: ObjectId('67cda32535e99230677ed8ee'),
  comp_name: 'Google',
  drive_date: 2020-11-15T00:00:00.000Z,
  user_id: ObjectId('67cd879635e99230677ed8d7')
}
{
  _id: ObjectId('67cda32535e99230677ed8ef'),
  comp_name: 'Microsoft',
  drive_date: 2020-12-10T00:00:00.000Z,
  user_id: ObjectId('67cd879635e99230677ed8d8')
}
```



Mentors Collection:

```
> db.mentors.insertMany([
  {
    mentor_name: "Thirupathy"
  },
  {
    mentor_name: "Manoj"
  },
  {
    mentor_name: "Palani"
  }
]);
```

```
> db.mentors.updateOne(
  { _id: ObjectId('67cda3f935e99230677ed8f2') },
  { $set: { mentees_count: 11 } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.mentors.find()
< {
  _id: ObjectId('67cda3f935e99230677ed8f0'),
  mentor_name: 'Thirupathy',
  mentees_count: 34
}
{
  _id: ObjectId('67cda3f935e99230677ed8f1'),
  mentor_name: 'Manoj',
  mentees_count: 8
}
{
  _id: ObjectId('67cda3f935e99230677ed8f2'),
  mentor_name: 'Palani',
  mentees_count: 11
}
```

Mentees count added using updateOne and \$set method

1) Find all the topics and tasks which are thought in the month of October

```
> db.topics.aggregate([
  {
    $match: {
      date: {
        $gte: new Date("2020-10-01T00:00:00.000Z"),
        $lte: new Date("2020-10-31T23:59:59.999Z")
      }
    }
  },
  {
    $lookup: {
      from: "tasks",
      localField: "_id",
      foreignField: "topic_id",
      as: "tasks"
    }
  }
]);
< {
  _id: ObjectId('67cd9e1335e99230677ed8df'),
  topic: 'HTML, CSS, JS',
  date: 2020-10-01T00:00:00.000Z,
  tasks: []
}
```

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

```
> db.company_drives.find({
  drive_date: {
    $gte: ISODate("2020-10-15"),
    $lte: ISODate("2020-10-31")
  }
});
< {
  _id: ObjectId('67cda32535e99230677ed8ed'),
  comp_name: 'Guvi',
  drive_date: 2020-10-20T00:00:00.000Z,
  user_id: ObjectId('67cd879635e99230677ed8d6')
}
```

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

```
> db.company_drives.aggregate([
  {
    $lookup: {
      from: "users",
      localField: "user_id",
      foreignField: "_id",
      as: "candidates"
    }
  },
  {
    $project: {
      _id: 0, comp_name: 1, drive_date: 1, candidates: 1
    }
  }
]);
< {
  comp_name: 'Guvi',
  drive_date: 2020-10-20T00:00:00.000Z,
  candidates: [
    {
      _id: ObjectId('67cd879635e99230677ed8d6'),
      name: 'Thiru',
      email: 'thiru@mail.com',
      batch: 'B17',
      task_id: ObjectId('67cd879635e99230677ed8cd'),
      mentor_id: ObjectId('67cd879635e99230677ed8ce'),
      codekata: ObjectId('67cd879635e99230677ed8cf'),
      attendance: 80
    }
  ]
}
{
  comp_name: 'Google',
  drive_date: 2020-11-15T00:00:00.000Z,
  candidates: [
```

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

```
> db.codekata.aggregate([
  { $group: { _id: '$user_id', total_solved: { $sum: "$solved" } } }
]);
< {
  _id: ObjectId('67cd879635e99230677ed8d8'),
  total_solved: 100
}
{
  _id: ObjectId('67cd879635e99230677ed8d7'),
  total_solved: 62
}
{
  _id: ObjectId('67cd879635e99230677ed8d6'),
  total_solved: 60
}
try>
```

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

```
> db.mentors.find({ mentees_count: { $gt: 15 } });  
< {  
  _id: ObjectId('67cda3f935e99230677ed8f0'),  
  mentor_name: 'Thirupathy',  
  mentees_count: 34  
}  
try>
```

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

Not resolved