GUVI Day 36 Task

This document is split into 2 sections.

1. Schema
2. Queries and Output

1. Schema:

For the given entities, Users, Mentors, Tasks, Attendance, code kata, company drives and topics, we draft a schema for each collection that is normalized and meaningful and helps us solve our questions.

a. **Users**:

```
{
    _id: ObjectID,
    name: "",
    email: "",
    username: "",
    password: "",
    mobile: "",
    dob: "",
    gender: "",
    years_of_experience: 0,
}
```

b. **Mentors**:

```
{
    _id: "",
    name: "",
    experience: 0,
    email: "",
    username: "",
    password: "",
    mobile: "",
    dob: "",
    gender: "",
    mentees: [], // List of User IDs as Foreign Key
}
```

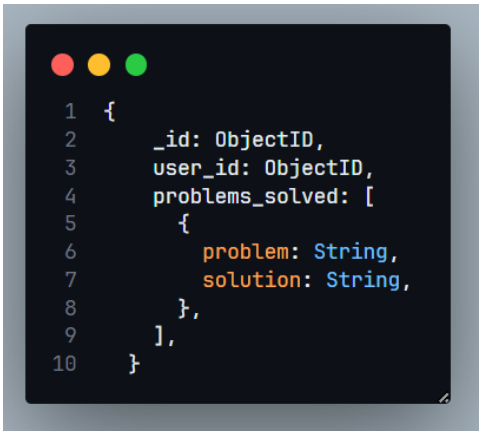
c. **Topics**:

```
{
    _id: ObjectID,
    title: String,
    description: String,
    date: Date,
  }
```

d. **Tasks**:

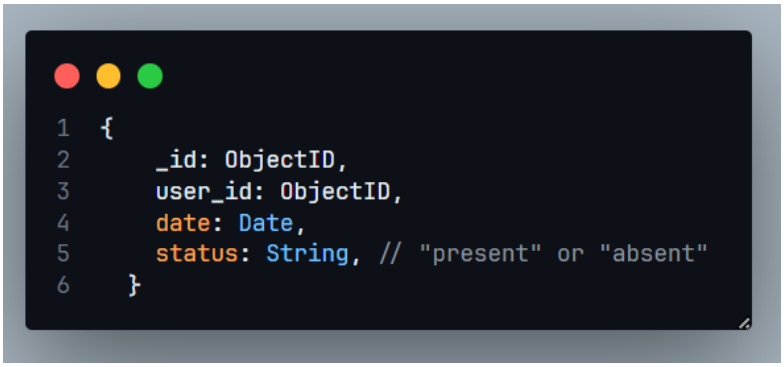
```
{
    _id: ObjectID,
    title: String,
    description: String,
    date: Date,
    submissions: [
      {
        user_id: ObjectID,
        status: String, // "submitted" or "not submitted"
      },
    ],
    topic: ObjectId,
  }
```

e. **Code kata**:



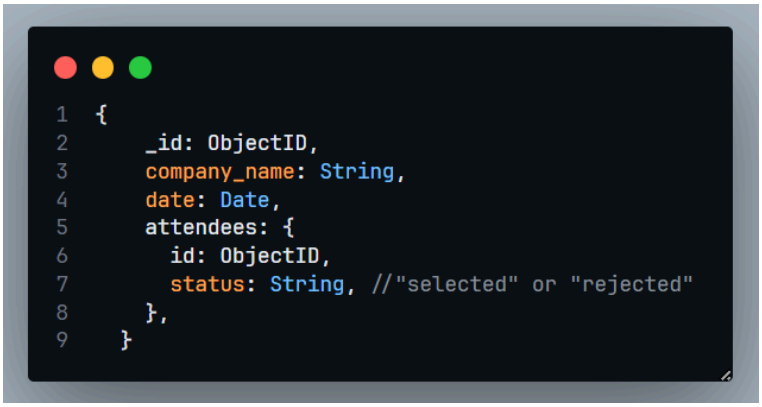
```
{
    _id: ObjectID,
    user_id: ObjectID,
    problems_solved: [
      {
        problem: String,
        solution: String,
      },
    ],
  }
```

f. **Attendance**:



```
{
    _id: ObjectID,
    user_id: ObjectID,
    date: Date,
    status: String, // "present" or "absent"
}
```

g. **Company Drive**:



```
{
    _id: ObjectID,
    company_name: String,
    date: Date,
    attendees: {
      id: ObjectID,
      status: String, //"selected" or "rejected"
    },
}
```

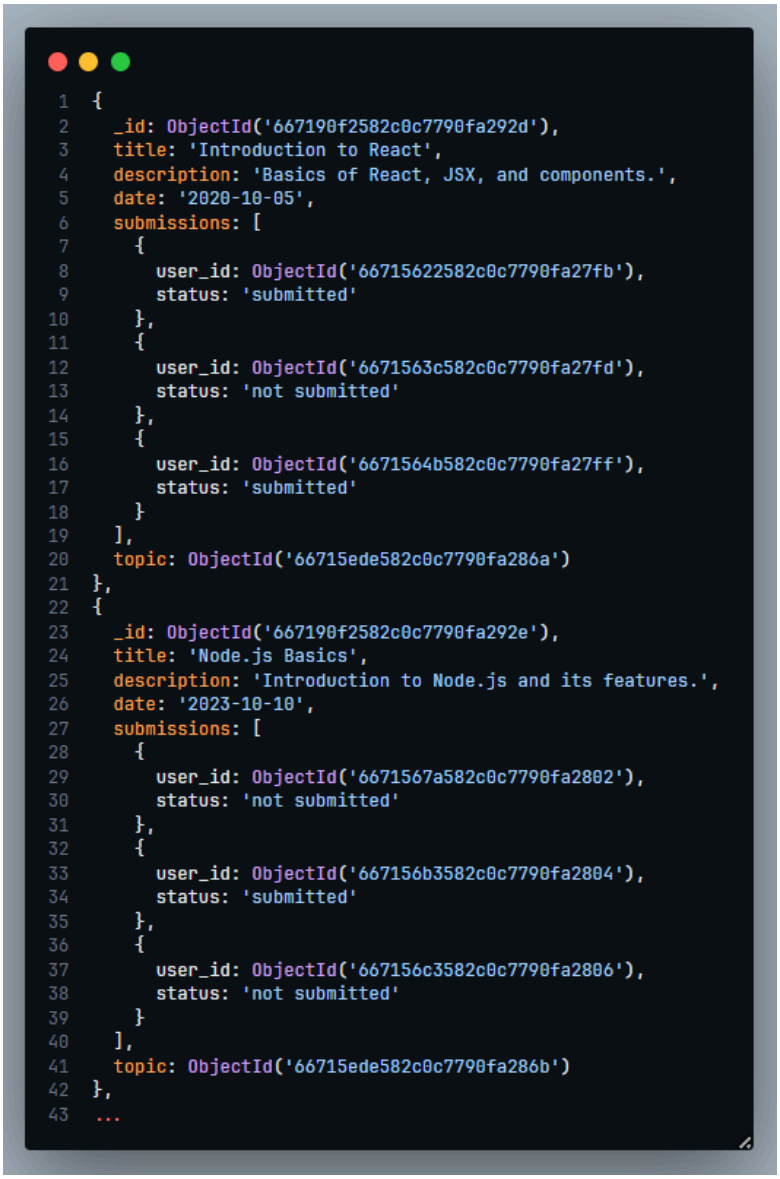
2. Queries and Output:

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

Query:

```
db.tasks.aggregate([
  {
    $match: {
      date: {
        $gte: "2020-10-01",
        $lt: "2024-11-01"
      }
    }
  }
]);
```

Solution:



```
{
  _id: ObjectId('667190f2582c0c7790fa292d'),
  title: 'Introduction to React',
  description: 'Basics of React, JSX, and components.',
  date: '2020-10-05',
  submissions: [
    {
      user_id: ObjectId('66715622582c0c7790fa27fb'),
      status: 'submitted'
    },
    {
      user_id: ObjectId('6671563c582c0c7790fa27fd'),
      status: 'not submitted'
    },
    {
      user_id: ObjectId('6671564b582c0c7790fa27ff'),
      status: 'submitted'
    }
  ],
  topic: ObjectId('66715ede582c0c7790fa286a')
},
{
  _id: ObjectId('667190f2582c0c7790fa292e'),
  title: 'Node.js Basics',
  description: 'Introduction to Node.js and its features.',
  date: '2023-10-10',
  submissions: [
    {
      user_id: ObjectId('6671567a582c0c7790fa2802'),
      status: 'not submitted'
    },
    {
      user_id: ObjectId('667156b3582c0c7790fa2804'),
      status: 'submitted'
    },
    {
      user_id: ObjectId('667156c3582c0c7790fa2806'),
      status: 'not submitted'
    }
  ],
  topic: ObjectId('66715ede582c0c7790fa286b')
},
...
```

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

Query:

```
db.company_drives.aggregate([
  {
    $match: {
      date: {
        $gte: "2020-10-15",
        $lt: "2020-10-31"
      }
    }
  }
]);
```

Solution:

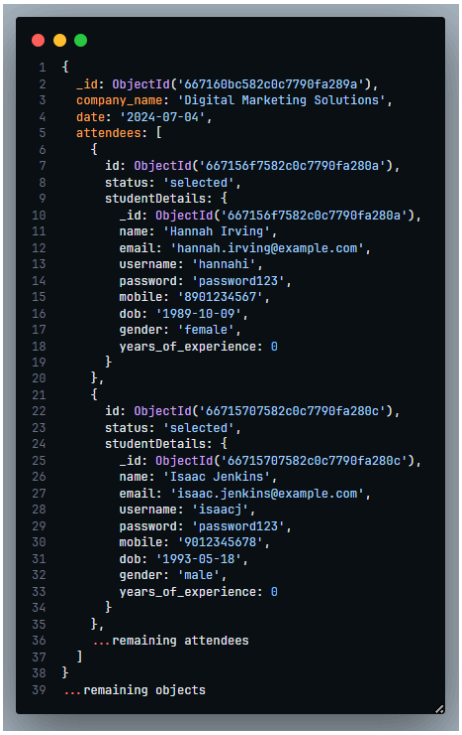
```
{
  _id: ObjectId('667165de582c0c7790fa28f3'),
  company_name: 'Innovative Web Solutions',
  date: '2020-10-23',
  attendees: [
    {
      id: ObjectId('66715716582c0c7790fa280e'),
      status: 'rejected'
    },
    {
      id: ObjectId('6671563c582c0c7790fa27fd'),
      status: 'selected'
    },
    {
      id: ObjectId('6671564b582c0c7790fa27ff'),
      status: 'selected'
    },
    {
      id: ObjectId('6671567a582c0c7790fa2802'),
      status: 'selected'
    },
    {
      id: ObjectId('667156b3582c0c7790fa2804'),
      status: 'rejected'
    }
  ]
}
...
```

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

Query:

```
db.company_drives.aggregate([
  {
    $unwind: "$attendees",
  },
  {
    $lookup: {
      from: "users",
      localField: "attendees.id",
      foreignField: "_id",
      as: "studentDetails",
    },
  },
  {
    $unwind: "$studentDetails",
  },
  {
    $group: {
      _id: "$_id",
      company_name: { $first: "$company_name" },
      date: { $first: "$date" },
      attendees: {
        $push: {
          id: "$attendees.id",
          status: "$attendees.status",
          studentDetails: "$studentDetails",
        },
      },
    },
  },
]);
```

Solution:



```
{
  _id: ObjectId('667160bc582c0c7790fa289a'),
  company_name: 'Digital Marketing Solutions',
  date: '2024-07-04',
  attendees: [
    {
      id: ObjectId('667156f7582c0c7790fa280a'),
      status: 'selected',
      studentDetails: {
        _id: ObjectId('667156f7582c0c7790fa280a'),
        name: 'Hannah Irving',
        email: 'hannah.irving@example.com',
        username: 'hannahi',
        password: 'password123',
        mobile: '8901234567',
        dob: '1989-10-09',
        gender: 'female',
        years_of_experience: 0
      }
    },
    {
      id: ObjectId('66715707582c0c7790fa280c'),
      status: 'selected',
      studentDetails: {
        _id: ObjectId('66715707582c0c7790fa280c'),
        name: 'Isaac Jenkins',
        email: 'isaac.jenkins@example.com',
        username: 'isaacj',
        password: 'password123',
        mobile: '9012345678',
        dob: '1993-05-18',
        gender: 'male',
        years_of_experience: 0
      }
    },
    ...remaining attendees
  ]
}
...remaining objects
```

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

Query:

```
db.code_kata.aggregate([
  {
      $match:{
          "user_id": ObjectId("66715716582c0c7790fa280e")
      }
  },
  {
  $project: {
    user_id: 1,
    problemsSolved: { $size: "$problems_solved" }
  }
}
])
```

Solution:

```
{
  _id: ObjectId('66715c2b582c0c7790fa284d'),
  user_id: ObjectId('66715716582c0c7790fa280e'),
  problemsSolved: 3
}
```

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

Query:

```
db.mentors.aggregate([
  {
    $project: {
      name: 1,
      menteeCount: { $size: "$mentees" }
    }
  },
  {
    $match: {
      menteeCount: { $gt: 15 }
    }
  }
]);
```

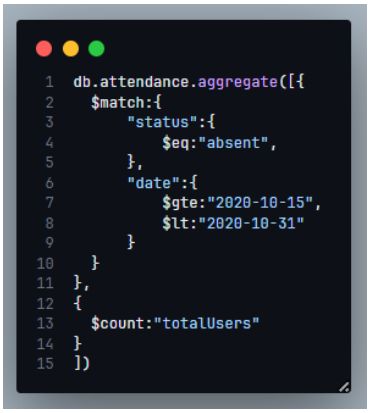
Solution:

```
{
  _id: ObjectId('6671688a582c0c7790fa2907'),
  name: 'Amit Kumar',
  menteeCount: 17
},
{
  _id: ObjectId('6671688a582c0c7790fa2908'),
  name: 'Priya Patel',
  menteeCount: 19
},
{
  _id: ObjectId('6671688a582c0c7790fa290a'),
  name: 'Rajesh Kumar',
  menteeCount: 18
}
```

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

For this problem, we have split the solution into 2 queries to get accurate results.

Query for absentees:



```
db.attendance.aggregate([{
  $match:{
      "status":{
          $eq:"absent",
      },
      "date":{
          $gte:"2020-10-15",
          $lt:"2020-10-31"
      }
  }
},
{
  $count:"totalUsers"
}
])
```

Solution:

```
{
  totalUsers: 4
}
```

Query for incomplete tasks:

```
db.tasks.aggregate([{
  $match:{
      "submissions.status":{
          $eq:"not submitted",
      },
      "date":{
          $gte:"2020-10-15",
          $lt:"2020-10-31"
      }
  }
},
{
  $count:"totalUsers"
}
])
```

Solution:

```
{
  totalUsers: 3
}
```