1. Users Collection

{

"\_id": ObjectId,

"username": String,

"email": String,

"password": String,

"created\_at": Date,

"updated\_at": Date,

"role": String // e.g., 'student', 'mentor', 'admin'

}

2. CodeKata Collection

{

"\_id": ObjectId,

"title": String,

"description": String,

"difficulty": String, // e.g., 'easy', 'medium', 'hard'

"created\_at": Date,

"updated\_at": Date

}

3. Attendance Collection

{

"\_id": ObjectId,

"user\_id": ObjectId, // Reference to Users

"date": Date,

"status": String, // e.g., 'present', 'absent'

"created\_at": Date,

"updated\_at": Date}

4. Topics Collection

{

"\_id": ObjectId,

"name": String,

"description": String,

"created\_at": Date,

"updated\_at": Date

}

5. **Tasks Collection**

{

"\_id": ObjectId,

"task\_title": String,

"description": String,

"due\_date": Date,

"assigned\_to": ObjectId, // Reference to Users

"status": String, // e.g., 'pending', 'completed'

"created\_at": Date,

"updated\_at": Date

}

6. Company Drives Collection

{

"\_id": ObjectId,

"drive\_name": String,

"description": String,

"link": String, // URL to the drive

"created\_at": Date,

"updated\_at": Date

}

7. Mentors Collection

{

"\_id": ObjectId,

"mentor\_id": ObjectId, // Reference to Users

"expertise": String,

"bio": String,

"availability": String, // e.g., 'available', 'busy'

"created\_at": Date,

"updated\_at": Date

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

db.topics.find({ created\_at: { $gte: new Date("2024-10-01"), $lt: new Date("2024-11-01") }})

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

db.company\_drives.find({

created\_at: {

$gte: new Date("2020-10-15"),

$lt: new Date("2020-11-01") // Exclude November 1st to include only October 31st

}

})

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

db.students.aggregate([

{

$match: { placed: true } // Filter students who have appeared for placement

},

{

$lookup: {

from: "company\_drives", // The collection to join

localField: "drive\_id", // Field from the students collection

foreignField: "\_id", // Field from the company\_drives collection

as: "company\_drive\_info" // Name of the new array field to add

}

},

{

$unwind: "$company\_drive\_info" // Deconstructs the array field from the lookup

},

{

$project: {

student\_name: "$name",

student\_email: "$email",

drive\_name: "$company\_drive\_info.drive\_name",

drive\_description: "$company\_drive\_info.description",

drive\_link: "$company\_drive\_info.link"

}

}])

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

const userId = ObjectId("user\_id\_here"); // Replace with the actual user ID

db.submissions.aggregate([

{

$match: {

user\_id: userId,

status: "solved" // Assuming you want only solved problems

}

},

{

$count: "solved\_problems" // Counts the number of documents

}

])

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

db.mentors.aggregate([

{

$match: {

$expr: {

$gt: [ { $size: "$mentees" }, 15 ] // Count of mentees greater than 15

}

}

},

{

$project: {

mentor\_name: "$name",

mentor\_email: "$email",

mentee\_count: { $size: "$mentees" } // Include the count of mentees in the output

}

}

])

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

db.users.aggregate([

{

$unwind: "$attendance" // Deconstruct attendance array

},

{

$match: {

"attendance.status": "absent",

"attendance.date": {

$gte: new Date("2020-10-15"),

$lt: new Date("2020-11-01") // Exclude November 1st to include only October 31st

}

}

},

{

$lookup: {

from: "tasks", // Join with tasks collection

localField: "\_id", // User ID from users collection

foreignField: "user\_id", // User ID in tasks collection

as: "submitted\_tasks" // Name of the new array field

}

},

{

$match: {

submitted\_tasks: { $eq: [] } // Ensure no tasks were submitted

}

},

{

$count: "absent\_users\_without\_tasks" // Count the number of users

}

])