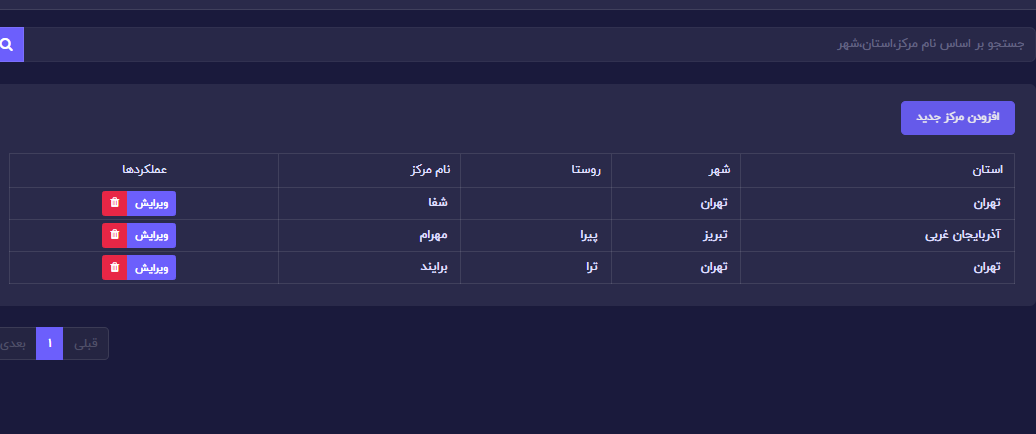


همونطور که تو تصویر میبینین برا من به ترتیب نشون میده و مرکز رو آخر نشون میده





اما اگر تو برا شما ترتیب درست نبود anydeskمیدین میام چک میکنم

خب در رابطه با توضیح بخش report

کد پاین کوئری اصلی هست که ما برای دریافت رکورد ها ازش استفاده کردیم

          patients = await Patient.aggregate([

این خط در ابتدا بعد از متغیر ما نوشتیم await این درواقع صبر میکنه تا کوئری که ما زدیم کامل اجرا بشه و در نهایت خروجی رو تو متغیر patients ذخیره میکنه

Patient در واقع مادلی هست برای کالکشن patients که در mongo وجود داره و از این مادل ما به همین کالکشن در دیتابیس دسترسی داریم

aggregate به معنی مجموع هست و درواقع برای اجرا کردن چند کوئری در کنار یکدیگر مورد استفاده قرار میگیره ، و در واقع یه آرایه به صورت pipeline میگیره و خروجی نهایی رو میده ، پایپ لاین هم از معنیش مشخص هست به معنی خط لوله یعنی اینطور در نظر داشته باشین که مثلا کوئری اول اجرا میشه یه خروجی میده کوئری بعدی روی اون خروجی کوئری اوله اجرا میشه و به همین ترتیب تا آخر .

                {

                    $match: {

                        "e\_crf.sideEffectDate": {

                            $gte: new Date(reportFrom),

                            $lt: new Date(reportTo)

                        }

                    }

                },

هر کدوم از کوئری های مجزا در داخل یک آبجکت قرار میگیرند و همه اونها داخل یک آرایه هستند []

بخش اول از کوئری ما در ابتدا نوشته شده $match این از نامش هست مشخص هست به معنی تطبیق هست و اون فیلدی که ما میخوایم رو با مقداری که قرار میدیم مطابقت میده و یه لیست از رکورد هایی که مطابقت دارند رو بر میگردونه.

فیلد مورد نظر ما هم که مشخص هست اونجا گفتیم تاریخ وقوع عارضه رو بررسی کنه ، $gte یعنی مقادیری که بیشتر از مقداری هستند که بهش میدیم رو در نظر بگیره.و $lt هم یعنی کمتر از این تایم درواقع به این شکل معنی میشه که تاریخ هایی که از تاریخ بالایی reportFromشروع میشن تا reportTo رو انتخاب کنه .

                { $unwind: '$e\_crf' },

Unwind معنی باز کردن رو میده ، درواقع ما یک فیلد از داکیومنت رو که یک آرایه هست بهش میدیم مقادیر مختلفی که اون آرایه گرفته رو بر میگردونه .مثلا همین '$e\_crf' یک آرایه از عارضه هایی است که اون بیمار داشته .یعنی مثلا

Name : ‘miveh’ noeMive:[‘sib’,’holo’,’gilas’]} {

این جا { $unwind: ‘$noeMive‘ },

این رو اگر بدیم خروجی که خواهیم داشت به این شکله

Name : ‘miveh’ noeMive:’sib’} {

Name : ‘miveh’ noeMive:’holo’} {

Name : ‘miveh’ noeMive:’gilas’} {

خب پس ما الان یه لیست آپدیت شده به دست آوردیم .که آرایه بیرونی همه بیمار هارو شامل میشه و آرایه داخلی هر بیمار رو به تعداد عارضه هایی که ثبت کرده و تو اون تاریخ وجود دارند به این شکل بالا لیست میکنه . بعد از این به کد پایین میرسیم.

                {

                    $match: {

                        "e\_crf.sideEffectDate": {

                            $gte: new Date(reportFrom),

                            $lt: new Date(reportTo)

                        }

                    }

                },

این کد نیز مشابه کد اولی هست با این تفاوت که همونطور که در بالا گفتم تو حالت فعلی خروجی و لیستی که ما به دست آوردیم آپدیت شده و ما رو اون لیست جدید کوئری میزنیم ، چرا؟ به این دلیل که چون داکیومنت ما از لیست های تودرتو استفاده میکنه برای اینکه بتونیم دقیقا اون چیزی که میخوایم رو از لیست های داخلی خارج کنیم به تنهایی با یک خروجی نمیشه و کل اون داکیومنت رو بر میگردونه مثلا اگه من بگم عارضه ای که تو تاریخ مثلا 7 تیر بوده رو برگردون خب اون داکیومنت بیمار رو بر میگردونه میگه بله این بیمار عارضه ای داره که تو هفت تیر رخ داده ، خود همون بیمار عارضه های دیگه ای هم داره و اونارو هم در واقع در کنار همین مورد تو خروجی میاره ، و خب ما نمیخوایم اینطوری باشه ما میخوایم دقیقا اونایی که تو 7 تیر بودن رو بیاره .

پس از همون unwind استفاده میکنیم تا لیست ها تکی بشن یعنی مثلا اونایی که مثل سیب هلو و گیلاس تو یه لیست بودن از هم جدا و هر کدوم یه آبجکت مجزا شده بودن روش کوئری میزنیم تا اینطوری دیگه دقیقا همون فیلد هارو میتونیم بگیریم پس به همین خاطر این کوئری بالارو دوباره تکرار کردیم اما با این تفاوت که دیگه این بار دقیقا همون فیلد هایی که اون تاریخ رو شامل میشن بگیریم .

در نهایت هم که عمل group کردن رو داریم که دیگه خودتون میدونین چی هست در واقع میایم گروه بندی میکنیم و فقط مقادیری که میخوایم رو میکشیم بیرون .

                {

                    $group: {

                        \_id: '$\_id',

                        list: { $push: '$e\_crf' },

                        userId: { $first: '$userId' },

                        birthday: { $first: '$birthday' },

                        gender: { $first: '$gender' },

                        educationRate: { $first: '$educationRate' },

                        jobStatus: { $first: '$jobStatus' },

                        ecoStatus: { $first: '$ecoStatus' },

                        phone: { $first: '$phone' },

                        address: { $first: '$address' },

                        user: { $first: '$user' }

                    }

                }

حالا چه مقادیری میخوایم من میگم آیدی بیمار رو بگیر ،

لیستی از عوارضی که داره رو همون e\_crf ها که قبلا یه لیست بودن و الان تکی هستن رو یکی یکی پوش کن به آرایه لیست ، list یه اسم و متغیر هست و هرچی میتونیم بزاریم کلا این بخش اول رو هر چی که بخوایم میتونیم اسم بدیم اما خب ما دقیقا همون اسم فیلد هارو دادیم + فقط e\_crf رو list دادیم که مجزا باشه و بدونیم چی به چی هست ، بقیه فیلد هارو هم میگیریم و قرار میدیم به این خاطر ما از $first استفاده کردیم چون لیست ما اگر بالا رو برید باز اون میوه هارو نگاه کنین نوشته name:’miveh’ این دیگه تو همشون یکسان هست و چون لیست ماننده از ما میخواد میگه خب کدوم رو میخوای من به عنوان مثلا gender این داکیومنت قرار بدم میگیم خب همون اولی قرار بده .

در نهایت یه لیست از اون عارضه هایی داریم که تو اون تاریخ رخ دادند .

اگر متوجه نشدید دوباره باز بخونین تا بتونین درکش کنین کمی پیچیده هست اما خب سعی کردم خوب توضیح بدم ، همچنین بقیه موارد هم دیگه چیزی نیستن تو ویدیو توضیح دادم که pdfچطوری گرفته میشه و

سعی کردم تو این فایل کوئری رو توضیح بدم براتون .

همچنین چند کوئری اصلی هم براتون خلاصه مانند توضیح میدم که هر جا دیدین بدونین چی هست .

const users = User.find();

لیست همه کاربران رو بر میگردونه

const users = User.find({username:’ali’});

اونهایی که یوزرنیمشون علی هست رو بر میگردونه

const users = User.find({username:’ali’,nCode:’1234’});

اونایی که یوزرنیم علی و کد ملی 1234 دارند بر میگردونه .

const users = await User.find({$or:[{username:’ali’},{username:’reza’}]});

اونایی که یوزرنیم علی یا رضا دارند رو بر میگردونه

const users = await User.findOneAndDelete({\_id:1});

اونی که آیدیش 1 هست رو پیدا و حذفش میکنه.

const users = await User.findOneAndUpdate({\_id:1},{fname:’reza’});

اونی که آیدی 1 داره رو پیدا و نامشو به رضا تغییر میده

const users = await User.create({fname:’reza’,username:’reza21’,nCode:’1201’});

یوزر جدیدی با نام رضا و یوزرنیم رضا21 و کدملی 1201 ایجاد میکنه .

            patients = await Patient.aggregate([

                {

                    $match: {

                        "e\_crf.sideEffectDate": {

                            $gte: new Date(reportFrom),

                            $lt: new Date(reportTo)

                        }

                    }

                },

                { $unwind: '$e\_crf' },

                {

                    $match: {

                        "e\_crf.sideEffectDate": {

                            $gte: new Date(reportFrom),

                            $lt: new Date(reportTo)

                        }

                    }

                },

                {

                    $group: {

                        \_id: '$\_id',

                        list: { $push: '$e\_crf' },

                        userId: { $first: '$userId' },

                        birthday: { $first: '$birthday' },

                        gender: { $first: '$gender' },

                        educationRate: { $first: '$educationRate' },

                        jobStatus: { $first: '$jobStatus' },

                        ecoStatus: { $first: '$ecoStatus' },

                        phone: { $first: '$phone' },

                        address: { $first: '$address' },

                        user: { $first: '$user' }

                    }

                }

            ]);