**Mern Stack**

**Name:** C Akil Shanmuga Sriram

**Unique ID:** E0222023

**Project Title:** Car Pooling Platform

**MongoDB Aggregation Operations for Car Pooling Platform**

**Users Collection Aggregations**

**Aggregation 1: User Statistics Dashboard**

**Purpose:** Generate comprehensive user statistics for admin dashboard  
 Use Case: Admin monitoring and platform analytics

javascript

db.users.aggregate([  
 {  
 $group: {  
 \_id: null,  
 totalUsers: { $sum: 1 },  
 activeUsers: {  
 $sum: { $cond: [{ $eq: ["$isActive", true] }, 1, 0] }  
 },  
 verifiedUsers: {  
 $sum: { $cond: [{ $eq: ["$profile.verified", true] }, 1, 0] }  
 },  
 driversOnly: {  
 $sum: { $cond: [{ $eq: ["$preferences.role", "driver"] }, 1, 0] }  
 },  
 ridersOnly: {  
 $sum: { $cond: [{ $eq: ["$preferences.role", "rider"] }, 1, 0] }  
 },  
 bothRoles: {  
 $sum: { $cond: [{ $eq: ["$preferences.role", "both"] }, 1, 0] }  
 },  
 avgRating: { $avg: "$rating.average" },  
 usersWithVehicles: {  
 $sum: { $cond: [{ $ne: ["$vehicle", null] }, 1, 0] }  
 }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 totalUsers: 1,  
 activeUsers: 1,  
 verifiedUsers: 1,  
 verificationRate: {  
 $multiply: [  
 { $divide: ["$verifiedUsers", "$totalUsers"] },  
 100  
 ]  
 },  
 roleDistribution: {  
 driversOnly: "$driversOnly",  
 ridersOnly: "$ridersOnly",  
 bothRoles: "$bothRoles"  
 },  
 avgRating: { $round: ["$avgRating", 2] },  
 usersWithVehicles: 1,  
 vehicleOwnershipRate: {  
 $multiply: [  
 { $divide: ["$usersWithVehicles", "$totalUsers"] },  
 100  
 ]  
 }  
 }  
 }  
]);

**View Name**: user\_statistics\_dashboard

**Aggregation 2: Top Rated Users by Location**

**Purpose**: Find highly rated users grouped by their location areas **Use Case**: Recommend trusted drivers/riders in specific areas

javascript

db.users.aggregate([  
 {  
 $match: {  
 "rating.totalRatings": { $gte: 5 },  
 "rating.average": { $gte: 4.0 },  
 isActive: true  
 }  
 },  
 {  
 $addFields: {  
 locationArea: {  
 $cond: {  
 if: { $regexMatch: { input: "$location.address", regex: /Gandhipuram/i } },  
 then: "Gandhipuram",  
 else: {  
 $cond: {  
 if: { $regexMatch: { input: "$location.address", regex: /RS Puram/i } },  
 then: "RS Puram",  
 else: {  
 $cond: {  
 if: { $regexMatch: { input: "$location.address", regex: /Saravanampatti/i } },  
 then: "Saravanampatti",  
 else: "Other Areas"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 {  
 $group: {  
 \_id: "$locationArea",  
 topUsers: {  
 $push: {  
 userId: "$\_id",  
 name: { $concat: ["$profile.firstName", " ", "$profile.lastName"] },  
 rating: "$rating.average",  
 totalRatings: "$rating.totalRatings",  
 role: "$preferences.role",  
 hasVehicle: { $cond: [{ $ne: ["$vehicle", null] }, true, false] }  
 }  
 },  
 avgAreaRating: { $avg: "$rating.average" },  
 userCount: { $sum: 1 }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 area: "$\_id",  
 topUsers: { $slice: [{ $sortArray: { input: "$topUsers", sortBy: { rating: -1 } } }, 10] },  
 avgAreaRating: { $round: ["$avgAreaRating", 2] },  
 userCount: 1  
 }  
 },  
 {  
 $sort: { avgAreaRating: -1 }  
 }  
]);

**View Name**: top\_rated\_users\_by\_location

**Aggregation 3: User Registration Trends**

**Purpose**: Analyze user registration patterns over time **Use Case**: Marketing insights and growth analysis

javascript

db.users.aggregate([  
 {  
 $group: {  
 \_id: {  
 year: { $year: "$createdAt" },  
 month: { $month: "$createdAt" }  
 },  
 newRegistrations: { $sum: 1 },  
 verifiedRegistrations: {  
 $sum: { $cond: [{ $eq: ["$profile.verified", true] }, 1, 0] }  
 },  
 driverRegistrations: {  
 $sum: { $cond: [{ $in: ["$preferences.role", ["driver", "both"]] }, 1, 0] }  
 }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 period: {  
 $concat: [  
 { $toString: "$\_id.year" },  
 "-",  
 { $toString: "$\_id.month" }  
 ]  
 },  
 year: "$\_id.year",  
 month: "$\_id.month",  
 newRegistrations: 1,  
 verifiedRegistrations: 1,  
 driverRegistrations: 1,  
 verificationRate: {  
 $multiply: [  
 { $divide: ["$verifiedRegistrations", "$newRegistrations"] },  
 100  
 ]  
 }  
 }  
 },  
 {  
 $sort: { year: 1, month: 1 }  
 }  
]);

**View Name**: user\_registration\_trends

**Rides Collection Aggregations**

**Aggregation 1: Popular Routes Analysis**

**Purpose**: Identify the most popular routes and their characteristics **Use Case**: Route optimization and demand analysis

javascript

db.rides.aggregate([  
 {  
 $match: {  
 status: { $in: ["active", "completed"] },  
 createdAt: { $gte: new Date(Date.now() - 30 \* 24 \* 60 \* 60 \* 1000) }  
 }  
 },  
 {  
 $addFields: {  
 routeKey: {  
 $concat: [  
 "$route.startLocation.address",  
 " -> ",  
 "$route.endLocation.address"  
 ]  
 }  
 }  
 },  
 {  
 $group: {  
 \_id: "$routeKey",  
 totalRides: { $sum: 1 },  
 avgCostPerSeat: { $avg: "$pricing.costPerSeat" },  
 avgDistance: { $avg: "$route.estimatedDistance" },  
 avgDuration: { $avg: "$route.estimatedDuration" },  
 totalSeatsOffered: { $sum: "$capacity.totalSeats" },  
 recurringRides: {  
 $sum: { $cond: ["$schedule.isRecurring", 1, 0] }  
 },  
 uniqueDrivers: { $addToSet: "$driverId" }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 route: "$\_id",  
 totalRides: 1,  
 avgCostPerSeat: { $round: ["$avgCostPerSeat", 2] },  
 avgDistance: { $round: ["$avgDistance", 1] },  
 avgDuration: { $round: ["$avgDuration", 0] },  
 totalSeatsOffered: 1,  
 recurringRides: 1,  
 uniqueDrivers: { $size: "$uniqueDrivers" },  
 costPerKm: {  
 $round: [  
 { $divide: ["$avgCostPerSeat", "$avgDistance"] },  
 2  
 ]  
 }  
 }  
 },  
 {  
 $sort: { totalRides: -1 }  
 },  
 {  
 $limit: 20  
 }  
]);

**View Name**: popular\_routes\_analysis

**Aggregation 2: Driver Performance Metrics**

**Purpose**: Evaluate driver performance and ride offerings **Use Case**: Driver ranking and performance monitoring

javascript

db.rides.aggregate([  
 {  
 $match: {  
 createdAt: { $gte: new Date(Date.now() - 30 \* 24 \* 60 \* 60 \* 1000) }  
 }  
 },  
 {  
 $group: {  
 \_id: "$driverId",  
 totalRidesOffered: { $sum: 1 },  
 completedRides: {  
 $sum: { $cond: [{ $eq: ["$status", "completed"] }, 1, 0] }  
 },  
 cancelledRides: {  
 $sum: { $cond: [{ $eq: ["$status", "cancelled"] }, 1, 0] }  
 },  
 totalSeatsOffered: { $sum: "$capacity.totalSeats" },  
 avgCostPerSeat: { $avg: "$pricing.costPerSeat" },  
 recurringRides: {  
 $sum: { $cond: ["$schedule.isRecurring", 1, 0] }  
 },  
 avgDistance: { $avg: "$route.estimatedDistance" }  
 }  
 },  
 {  
 $lookup: {  
 from: "users",  
 localField: "\_id",  
 foreignField: "\_id",  
 as: "driverInfo"  
 }  
 },  
 {  
 $unwind: "$driverInfo"  
 },  
 {  
 $project: {  
 \_id: 0,  
 driverId: "$\_id",  
 driverName: {  
 $concat: ["$driverInfo.profile.firstName", " ", "$driverInfo.profile.lastName"]  
 },  
 driverRating: "$driverInfo.rating.average",  
 totalRidesOffered: 1,  
 completedRides: 1,  
 cancelledRides: 1,  
 completionRate: {  
 $multiply: [  
 { $divide: ["$completedRides", "$totalRidesOffered"] },  
 100  
 ]  
 },  
 cancellationRate: {  
 $multiply: [  
 { $divide: ["$cancelledRides", "$totalRidesOffered"] },  
 100  
 ]  
 },  
 totalSeatsOffered: 1,  
 avgCostPerSeat: { $round: ["$avgCostPerSeat", 2] },  
 recurringRides: 1,  
 avgDistance: { $round: ["$avgDistance", 1] }  
 }  
 },  
 {  
 $sort: { completionRate: -1, totalRidesOffered: -1 }  
 }  
]);

**View Name**: driver\_performance\_metrics

**Bookings Collection Aggregations**

**Aggregation 1: Booking Success Rate Analysis**

**Purpose**: Analyze booking conversion rates and success patterns **Use Case**: Improve booking process and user experience

javascript

db.bookings.aggregate([  
 {  
 $group: {  
 \_id: {  
 year: { $year: "$requestedAt" },  
 month: { $month: "$requestedAt" }

},  
 totalBookings: { $sum: 1 },  
 confirmedBookings: {  
 $sum: { $cond: [{ $eq: ["$status", "confirmed"] }, 1, 0] }  
 },  
 completedBookings: {  
 $sum: { $cond: [{ $eq: ["$status", "completed"] }, 1, 0] }  
 },  
 cancelledBookings: {  
 $sum: { $cond: [{ $eq: ["$status", "cancelled"] }, 1, 0] }  
 },  
 pendingBookings: {  
 $sum: { $cond: [{ $eq: ["$status", "requested"] }, 1, 0] }  
 },  
 totalRevenue: {  
 $sum: {  
 $cond: [  
 { $eq: ["$status", "completed"] },  
 "$totalCost",  
 0  
 ]  
 }  
 },  
 avgBookingValue: { $avg: "$totalCost" },  
 totalSeatsBooked: { $sum: "$seatsBooked" }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 period: {  
 $concat: [  
 { $toString: "$\_id.year" },  
 "-",  
 { $toString: "$\_id.month" }  
 ]  
 },  
 totalBookings: 1,  
 confirmedBookings: 1,  
 completedBookings: 1,  
 cancelledBookings: 1,  
 pendingBookings: 1,  
 confirmationRate: {  
 $multiply: [  
 { $divide: ["$confirmedBookings", "$totalBookings"] },  
 100  
 ]  
 },  
 completionRate: {  
 $multiply: [  
 { $divide: ["$completedBookings", "$confirmedBookings"] },  
 100  
 ]  
 },  
 cancellationRate: {  
 $multiply: [  
 { $divide: ["$cancelledBookings", "$totalBookings"] },  
 100  
 ]  
 },  
 totalRevenue: { $round: ["$totalRevenue", 2] },  
 avgBookingValue: { $round: ["$avgBookingValue", 2] },  
 totalSeatsBooked: 1  
 }  
 },  
 {  
 $sort: { "\_id.year": 1, "\_id.month": 1 }  
 }  
]);

**View Name**: booking\_success\_rate\_analysis

**Aggregation 2: User Booking Behavior Patterns**

**Purpose**: Analyze individual user booking patterns and preferences **Use Case**: Personalized recommendations and user segmentation

javascript

db.bookings.aggregate([  
 {  
 $match: {  
 requestedAt: { $gte: new Date(Date.now() - 90 \* 24 \* 60 \* 60 \* 1000) }  
 }  
 },  
 {  
 $group: {  
 \_id: "$riderId",  
 totalBookings: { $sum: 1 },  
 completedBookings: {  
 $sum: { $cond: [{ $eq: ["$status", "completed"] }, 1, 0] }  
 },  
 cancelledBookings: {  
 $sum: { $cond: [{ $eq: ["$status", "cancelled"] }, 1, 0] }  
 },  
 totalSpent: {  
 $sum: {  
 $cond: [  
 { $eq: ["$paymentStatus", "completed"] },  
 "$totalCost",  
 0  
 ]  
 }  
 },  
 avgBookingValue: { $avg: "$totalCost" },  
 totalSeatsBooked: { $sum: "$seatsBooked" },  
 uniqueDrivers: { $addToSet: "$driverId" },  
 lastBookingDate: { $max: "$requestedAt" },  
 firstBookingDate: { $min: "$requestedAt" },  
 avgSeatsPerBooking: { $avg: "$seatsBooked" }  
 }  
 },  
 {  
 $lookup: {  
 from: "users",  
 localField: "\_id",  
 foreignField: "\_id",  
 as: "userInfo"  
 }  
 },  
 {  
 $unwind: "$userInfo"  
 },  
 {  
 $project: {  
 \_id: 0,  
 riderId: "$\_id",  
 riderName: {  
 $concat: ["$userInfo.profile.firstName", " ", "$userInfo.profile.lastName"]  
 },  
 userRating: "$userInfo.rating.average",  
 totalBookings: 1,  
 completedBookings: 1,  
 cancelledBookings: 1,  
 completionRate: {  
 $multiply: [  
 { $divide: ["$completedBookings", "$totalBookings"] },  
 100  
 ]  
 },  
 cancellationRate: {  
 $multiply: [  
 { $divide: ["$cancelledBookings", "$totalBookings"] },  
 100  
 ]  
 },  
 totalSpent: { $round: ["$totalSpent", 2] },  
 avgBookingValue: { $round: ["$avgBookingValue", 2] },  
 totalSeatsBooked: 1,  
 uniqueDriversUsed: { $size: "$uniqueDrivers" },  
 avgSeatsPerBooking: { $round: ["$avgSeatsPerBooking", 1] },  
 daysSinceLastBooking: {  
 $divide: [  
 { $subtract: [new Date(), "$lastBookingDate"] },  
 86400000  
 ]  
 },  
 userSegment: {  
 $switch: {  
 branches: [  
 { case: { $gte: ["$totalBookings", 20] }, then: "Power User" },  
 { case: { $gte: ["$totalBookings", 10] }, then: "Regular User" },  
 { case: { $gte: ["$totalBookings", 5] }, then: "Occasional User" },  
 { case: { $lt: ["$totalBookings", 5] }, then: "New User" }  
 ]  
 }  
 }  
 }  
 },  
 {  
 $sort: { totalSpent: -1 }  
 }  
]);

**View Name**: user\_booking\_behavior\_patterns

**Messages Collection Aggregations**

**Aggregation 1: Communication Activity Dashboard**

**Purpose**: Monitor messaging activity and user engagement **Use Case**: Platform engagement metrics and user support insights

javascript

db.messages.aggregate([  
 {  
 $match: {  
 sentAt: { $gte: new Date(Date.now() - 30 \* 24 \* 60 \* 60 \* 1000) }  
 }  
 },  
 {  
 $group: {  
 \_id: {  
 year: { $year: "$sentAt" },  
 month: { $month: "$sentAt" },  
 day: { $dayOfMonth: "$sentAt" }  
 },  
 totalMessages: { $sum: 1 },  
 uniqueSenders: { $addToSet: "$senderId" },  
 uniqueReceivers: { $addToSet: "$receiverId" },  
 uniqueConversations: { $addToSet: "$conversationId" },  
 systemMessages: {  
 $sum: { $cond: [{ $eq: ["$messageType", "system"] }, 1, 0] }  
 },  
 textMessages: {  
 $sum: { $cond: [{ $eq: ["$messageType", "text"] }, 1, 0] }  
 },  
 readMessages: {  
 $sum: { $cond: ["$isRead", 1, 0] }  
 },  
 avgMessageLength: { $avg: { $strLenCP: "$content" } }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 date: {  
 $dateFromParts: {  
 year: "$\_id.year",  
 month: "$\_id.month",  
 day: "$\_id.day"  
 }  
 },  
 totalMessages: 1,  
 uniqueSenders: { $size: "$uniqueSenders" },  
 uniqueReceivers: { $size: "$uniqueReceivers" },  
 uniqueConversations: { $size: "$uniqueConversations" },  
 systemMessages: 1,  
 textMessages: 1,  
 readMessages: 1,  
 readRate: {  
 $multiply: [  
 { $divide: ["$readMessages", "$totalMessages"] },  
 100  
 ]  
 },  
 avgMessageLength: { $round: ["$avgMessageLength", 0] },  
 messagesPerConversation: {  
 $round: [  
 { $divide: ["$totalMessages", { $size: "$uniqueConversations" }] },  
 1  
 ]  
 }  
 }  
 },  
 {  
 $sort: { date: 1 }  
 }  
]);

**View Name**: communication\_activity\_dashboard

**Ratings Collection Aggregations**

**Aggregation 1: Rating Distribution Analysis**

**Purpose**: Analyze rating distribution across platform users **Use Case**: Quality assessment and user satisfaction monitoring

javascript

db.ratings.aggregate([  
 {  
 $group: {  
 \_id: "$rating",  
 count: { $sum: 1 },  
 avgPunctuality: { $avg: "$categories.punctuality" },  
 avgCleanliness: { $avg: "$categories.cleanliness" },  
 avgCommunication: { $avg: "$categories.communication" },  
 avgSafety: { $avg: "$categories.safety" },  
 driverRatings: {  
 $sum: { $cond: [{ $eq: ["$ratingType", "driver"] }, 1, 0] }  
 },  
 riderRatings: {  
 $sum: { $cond: [{ $eq: ["$ratingType", "rider"] }, 1, 0] }  
 },  
 anonymousRatings: {  
 $sum: { $cond: ["$isAnonymous", 1, 0] }  
 },  
 ratingsWithReviews: {  
 $sum: { $cond: [{ $ne: ["$review", ""] }, 1, 0] }  
 }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 rating: "$\_id",  
 count: 1,  
 avgPunctuality: { $round: ["$avgPunctuality", 2] },  
 avgCleanliness: { $round: ["$avgCleanliness", 2] },  
 avgCommunication: { $round: ["$avgCommunication", 2] },  
 avgSafety: { $round: ["$avgSafety", 2] },  
 driverRatings: 1,  
 riderRatings: 1,  
 anonymousRatings: 1,  
 ratingsWithReviews: 1,  
 reviewRate: {  
 $multiply: [  
 { $divide: ["$ratingsWithReviews", "$count"] },  
 100  
 ]  
 }  
 }  
 },  
 {  
 $sort: { rating: -1 }  
 }  
]);

**View Name**: rating\_distribution\_analysis

**Payments Collection Aggregations**

**Aggregation 1: Financial Performance Dashboard**

**Purpose**: Track financial metrics and payment trends **Use Case**: Revenue analysis and financial reporting

javascript

db.payments.aggregate([  
 {  
 $match: {  
 createdAt: { $gte: new Date(Date.now() - 90 \* 24 \* 60 \* 60 \* 1000) }  
 }  
 },  
 {  
 $group: {  
 \_id: {  
 year: { $year: "$createdAt" },  
 month: { $month: "$createdAt" },  
 week: { $week: "$createdAt" }  
 },  
 totalTransactions: { $sum: 1 },  
 completedTransactions: {  
 $sum: { $cond: [{ $eq: ["$status", "completed"] }, 1, 0] }  
 },  
 pendingTransactions: {  
 $sum: { $cond: [{ $eq: ["$status", "pending"] }, 1, 0] }  
 },  
 failedTransactions: {  
 $sum: { $cond: [{ $eq: ["$status", "failed"] }, 1, 0] }  
 },  
 refundedTransactions: {  
 $sum: { $cond: [{ $eq: ["$status", "refunded"] }, 1, 0] }  
 },  
 totalRevenue: {  
 $sum: {  
 $cond: [{ $eq: ["$status", "completed"] }, "$amount", 0]  
 }  
 },  
 totalRefunds: { $sum: "$refundAmount" },  
 avgTransactionAmount: { $avg: "$amount" }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 period: {  
 $concat: [  
 { $toString: "$\_id.year" },  
 "-W",  
 { $toString: "$\_id.week" }  
 ]  
 },  
 year: "$\_id.year",  
 month: "$\_id.month",  
 week: "$\_id.week",  
 totalTransactions: 1,  
 completedTransactions: 1,  
 pendingTransactions: 1,  
 failedTransactions: 1,  
 refundedTransactions: 1,  
 successRate: {  
 $multiply: [  
 { $divide: ["$completedTransactions", "$totalTransactions"] },  
 100  
 ]  
 },  
 failureRate: {  
 $multiply: [  
 { $divide: ["$failedTransactions", "$totalTransactions"] },  
 100  
 ]  
 },  
 totalRevenue: { $round: ["$totalRevenue", 2] },  
 netRevenue: {  
 $round: [{ $subtract: ["$totalRevenue", "$totalRefunds"] }, 2]  
 },  
 totalRefunds: { $round: ["$totalRefunds", 2] },  
 avgTransactionAmount: { $round: ["$avgTransactionAmount", 2] }  
 }  
 },  
 {  
 $sort: { year: 1, week: 1 }  
 }  
]);

**View Name**: financial\_performance\_dashboard

**Safety Reports Collection Aggregations**

**Aggregation 1: Safety Incidents Dashboard**

**Purpose**: Monitor safety incidents and resolution patterns **Use Case**: Safety management and platform security monitoring

javascript

db.safety\_reports.aggregate([  
 {  
 $group: {  
 \_id: {  
 year: { $year: "$createdAt" },  
 month: { $month: "$createdAt" }  
 },  
 totalReports: { $sum: 1 },  
 openReports: {  
 $sum: { $cond: [{ $eq: ["$status", "open"] }, 1, 0] }  
 },  
 investigatingReports: {  
 $sum: { $cond: [{ $eq: ["$status", "investigating"] }, 1, 0] }  
 },  
 resolvedReports: {  
 $sum: { $cond: [{ $eq: ["$status", "resolved"] }, 1, 0] }  
 },  
 closedReports: {  
 $sum: { $cond: [{ $eq: ["$status", "closed"] }, 1, 0] }  
 },  
 criticalReports: {  
 $sum: { $cond: [{ $eq: ["$priority", "critical"] }, 1, 0] }  
 },  
 highPriorityReports: {  
 $sum: { $cond: [{ $eq: ["$priority", "high"] }, 1, 0] }  
 },  
 avgResolutionTime: {  
 $avg: {  
 $cond: [  
 { $ne: ["$resolvedAt", null] },  
 { $subtract: ["$resolvedAt", "$createdAt"] },  
 null  
 ]  
 }  
 },  
 uniqueReporters: { $addToSet: "$reporterId" },  
 uniqueReportedUsers: { $addToSet: "$reportedUserId" }  
 }  
 },  
 {  
 $project: {  
 \_id: 0,  
 period: {  
 $concat: [  
 { $toString: "$\_id.year" },  
 "-",  
 { $toString: "$\_id.month" }  
 ]  
 },  
 totalReports: 1,  
 openReports: 1,  
 investigatingReports: 1,  
 resolvedReports: 1,  
 closedReports: 1,  
 resolutionRate: {  
 $multiply: [  
 {  
 $divide: [  
 { $add: ["$resolvedReports", "$closedReports"] },  
 "$totalReports"  
 ]  
 },  
 100  
 ]  
 },  
 criticalReports: 1,  
 highPriorityReports: 1,  
 urgentReportsRate: {  
 $multiply: [  
 {  
 $divide: [  
 { $add: ["$criticalReports", "$highPriorityReports"] },  
 "$totalReports"  
 ]  
 },  
 100  
 ]  
 },  
 avgResolutionTimeDays: {  
 $round: [  
 { $divide: ["$avgResolutionTime", 86400000] },  
 1  
 ]  
 },  
 uniqueReporters: { $size: "$uniqueReporters" },  
 uniqueReportedUsers: { $size: "$uniqueReportedUsers" }  
 }  
 },  
 {  
 $sort: { "\_id.year": 1, "\_id.month": 1 }  
 }  
]);

**View Name**: safety\_incidents\_dashboard

**Aggregation 2: High-Risk Users Identification**

**Purpose**: Identify users with multiple safety reports **Use Case**: User monitoring and risk assessment

javascript

db.safety\_reports.aggregate([  
 {  
 $match: {  
 createdAt: { $gte: new Date(Date.now() - 180 \* 24 \* 60 \* 60 \* 1000) }  
 }  
 },  
 {  
 $group: {  
 \_id: "$reportedUserId",  
 totalReportsAgainst: { $sum: 1 },  
 incidentTypes: { $push: "$incidentType" },  
 priorities: { $push: "$priority" },  
 statuses: { $push: "$status" },  
 latestIncident: { $max: "$createdAt" },  
 earliestIncident: { $min: "$createdAt" },  
 resolvedIncidents: {  
 $sum: { $cond: [{ $eq: ["$status", "resolved"] }, 1, 0] }  
 },  
 openIncidents: {  
 $sum: { $cond: [{ $eq: ["$status", "open"] }, 1, 0] }  
 },  
 criticalIncidents: {  
 $sum: { $cond: [{ $eq: ["$priority", "critical"] }, 1, 0] }  
 },  
 uniqueReporters: { $addToSet: "$reporterId" }  
 }  
 },  
 {  
 $match: {  
 totalReportsAgainst: { $gte: 2 }  
 }  
 },  
 {  
 $lookup: {  
 from: "users",  
 localField: "\_id",  
 foreignField: "\_id",  
 as: "userInfo"  
 }  
 },  
 {  
 $unwind: "$userInfo"  
 },  
 {  
 $project: {  
 \_id: 0,  
 userId: "$\_id",  
 userName: {  
 $concat: ["$userInfo.profile.firstName", " ", "$userInfo.profile.lastName"]  
 },  
 userRole: "$userInfo.preferences.role",  
 userRating: "$userInfo.rating.average",  
 totalReportsAgainst: 1,  
 resolvedIncidents: 1,  
 openIncidents: 1,  
 criticalIncidents: 1,

resolutionRate: {  
 $multiply: [  
 { $divide: ["$resolvedIncidents", "$totalReportsAgainst"] },  
 100  
 ]  
 },  
 daysSinceLatestIncident: {  
 $round: [  
 { $divide: [{ $subtract: [new Date(), "$latestIncident"] }, 86400000] },  
 0  
 ]  
 },  
 uniqueReporters: { $size: "$uniqueReporters" },  
 riskScore: {  
 $add: [  
 { $multiply: ["$totalReportsAgainst", 10] },  
 { $multiply: ["$criticalIncidents", 25] },  
 { $multiply: ["$openIncidents", 15] },  
 {  
 $multiply: [  
 {  
 $subtract: [  
 100,  
 { $multiply: [{ $divide: ["$resolvedIncidents", "$totalReportsAgainst"] }, 100] }  
 ]  
 },  
 0.5  
 ]  
 }  
 ]  
 },  
 riskLevel: {  
 $switch: {  
 branches: [  
 { case: { $gte: ["$totalReportsAgainst", 5] }, then: "High Risk" },  
 { case: { $gte: ["$totalReportsAgainst", 3] }, then: "Medium Risk" },  
 { case: { $gte: ["$totalReportsAgainst", 2] }, then: "Low Risk" }  
 ]  
 }  
 }  
 }  
 },  
 {  
 $sort: { riskScore: -1, totalReportsAgainst: -1 }  
 }  
]);

**View Name**: high\_risk\_users\_identification

**Aggregation Views Creation Commands**

To create the aggregation views in MongoDB Atlas, use the following commands:

javascript

*// Users Collection Views*  
db.createView("user\_statistics\_dashboard", "users", [  
*// Insert the user statistics aggregation pipeline here*  
]);  
  
db.createView("top\_rated\_users\_by\_location", "users", [  
 *// Insert the top rated users aggregation pipeline here*  
]);  
  
db.createView("user\_registration\_trends", "users", [  
 *// Insert the registration trends aggregation pipeline here*  
]);  
  
*// Rides Collection Views*  
db.createView("popular\_routes\_analysis", "rides", [  
 *// Insert the popular routes aggregation pipeline here*  
]);  
  
db.createView("driver\_performance\_metrics", "rides", [  
 *// Insert the driver performance aggregation pipeline here*  
]);  
  
*// Bookings Collection Views*  
db.createView("booking\_success\_rate\_analysis", "bookings", [  
 *// Insert the booking success rate aggregation pipeline here*  
]);  
  
db.createView("user\_booking\_behavior\_patterns", "bookings", [  
 *// Insert the user booking behavior aggregation pipeline here*  
]);  
  
*// Messages Collection Views*  
db.createView("communication\_activity\_dashboard", "messages", [  
 *// Insert the communication activity aggregation pipeline here*  
]);  
  
*// Ratings Collection Views*  
db.createView("rating\_distribution\_analysis", "ratings", [  
 *// Insert the rating distribution aggregation pipeline here*  
]);  
  
*// Payments Collection Views*  
db.createView("financial\_performance\_dashboard", "payments", [  
 *// Insert the financial performance aggregation pipeline here*  
]);  
  
*// Safety Reports Collection Views*  
db.createView("safety\_incidents\_dashboard", "safety\_reports", [  
 *// Insert the safety incidents aggregation pipeline here*  
]);  
  
db.createView("high\_risk\_users\_identification", "safety\_reports", [  
 *// Insert the high risk users aggregation pipeline here*  
]);

**Why Each Aggregation Was Used**

**Users Collection Aggregations:**

1. **User Statistics Dashboard**: Provides comprehensive overview of user base for admin monitoring and business intelligence
2. **Top Rated Users by Location**: Helps recommend trusted drivers/riders in specific areas, improving match quality
3. **User Registration Trends**: Analyzes growth patterns to inform marketing strategies and resource allocation

**Rides Collection Aggregations:**

1. **Popular Routes Analysis**: Identifies high-demand routes for strategic planning and resource optimization
2. **Driver Performance Metrics**: Evaluates driver reliability and service quality for platform improvement

**Bookings Collection Aggregations:**

1. **Booking Success Rate Analysis**: Tracks conversion metrics to identify bottlenecks in the booking process
2. **User Booking Behavior Patterns**: Enables personalized recommendations and targeted user engagement

**Messages Collection Aggregations:**

1. **Communication Activity Dashboard**: Monitors user engagement and platform usage patterns

**Ratings Collection Aggregations:**

1. **Rating Distribution Analysis**: Assesses overall service quality and user satisfaction levels

**Payments Collection Aggregations:**

1. **Financial Performance Dashboard**: Tracks revenue, transaction success rates, and financial health

**Safety Reports Collection Aggregations:**

1. **Safety Incidents Dashboard**: Monitors platform safety and incident resolution efficiency
2. **High-Risk Users Identification**: Proactively identifies problematic users for safety management

**Business Value of Aggregations**

These aggregation operations provide:

**Operational Insights:**

* Real-time monitoring of platform health
* Performance tracking of drivers and riders
* Safety incident management
* Financial performance analysis

**Strategic Intelligence:**

* User growth and retention patterns
* Popular route identification for expansion
* Payment method preferences and success rates
* Geographic distribution of high-quality users

**Quality Assurance:**

* Rating distribution analysis for service quality
* Communication pattern monitoring for user engagement
* Safety report tracking for platform security
* User risk assessment for proactive management