

MongoDB Event Management System

Name: Priyanshi Bisht

Subject: MongoDB Assignment

Topic: Event, Ticket & User Management System

1. Introduction

This project is designed to demonstrate the use of MongoDB for managing an Event Management System. The system allows users to browse events, book tickets, and organizers to manage events and track ticket sales.

2. Database Schema Design

The database consists of four main collections: Users, Events, Tickets, and Categories. Each collection is structured to support efficient data storage and querying.

Users: Stores attendee and organizer details.

Events: Stores event details created by organizers.

Tickets: Stores ticket booking information.

Categories: Stores event categories like Music, Tech, Sports.

3. Users Collection

The Users collection stores information about attendees and organizers. Each user has a role that defines their access in the system.

The screenshot shows the MongoDB Compass interface connected to the 'eventManagementDB' database. The 'categories' collection is selected, displaying 20 documents. Each document contains an '_id' field (ObjectID), a 'name' field (String), and a 'description' field (String). The categories listed are: Arts, Business, Education, Health & wellness, Dance, Dance shows, Stage performances, Startup, AI, Machine Learning, Coding, Coding workshops, Finance, Marketing, Design, Photography, Gaming, Gaming tournaments, Yoga, and Meditation. The interface includes a sidebar for connections and a toolbar with various operations like ADD DATA, UPDATE, DELETE, EXPORT DATA, and EXPORT CODE.

_id	name	description
4	"Arts"	"ART EXHIBITIONS"
5	"Business"	"Business events"
6	"Education"	"Educational seminars"
7	"Health"	"Health & wellness"
8	"Dance"	"Dance shows"
9	"Theatre"	"Stage performances"
10	"Startup"	"Startup events"
11	"AI"	"AI conferences"
12	"ML"	"Machine Learning"
13	"Coding"	"Coding workshops"
14	"Finance"	"Finance summits"
15	"Marketing"	"Marketing events"
16	"Design"	"Design talks"
17	"Photography"	"Photo events"
18	"Gaming"	"Gaming tournaments"
19	"Yoga"	"Yoga sessions"
20	"Meditation"	"Mindfulness events"

4. Events Collection

The Events collection stores event details such as title, date, venue, price, and available tickets. Each event is linked to an organizer.

This screenshot shows the MongoDB Compass interface for the 'events' collection. The interface includes a sidebar for 'Connections' and 'Data Modeling', and a main area for querying and managing documents. Three documents are listed:

```
_id: ObjectId("696de24e6d56c739c768e2e5")
title: "Rock Music Night"
description: "Live rock concert"
categoryId: ObjectId("696de1eb6d56c739c768e2e1")
dateTime: 2026-03-28T08:00:00+00:00
venue: "No idea"
organizerId: ObjectId("696de0a95a8b2c5e06818b80")
price: 1500
totalTickets: 300
availableTickets: 200
status: "Completed"

_id: ObjectId("696de24e6d56c739c768e2e6")
title: "Tech Summit 2026"
description: "AI & Web Conference"
categoryId: ObjectId("696de1eb6d56c739c768e2e2")
dateTime: 2026-04-10T08:00:00+00:00
venue: "No idea"
organizerId: ObjectId("696de0a95a8b2c5e06818b82")
price: 1000
totalTickets: 400
availableTickets: 250
status: "Upcoming"

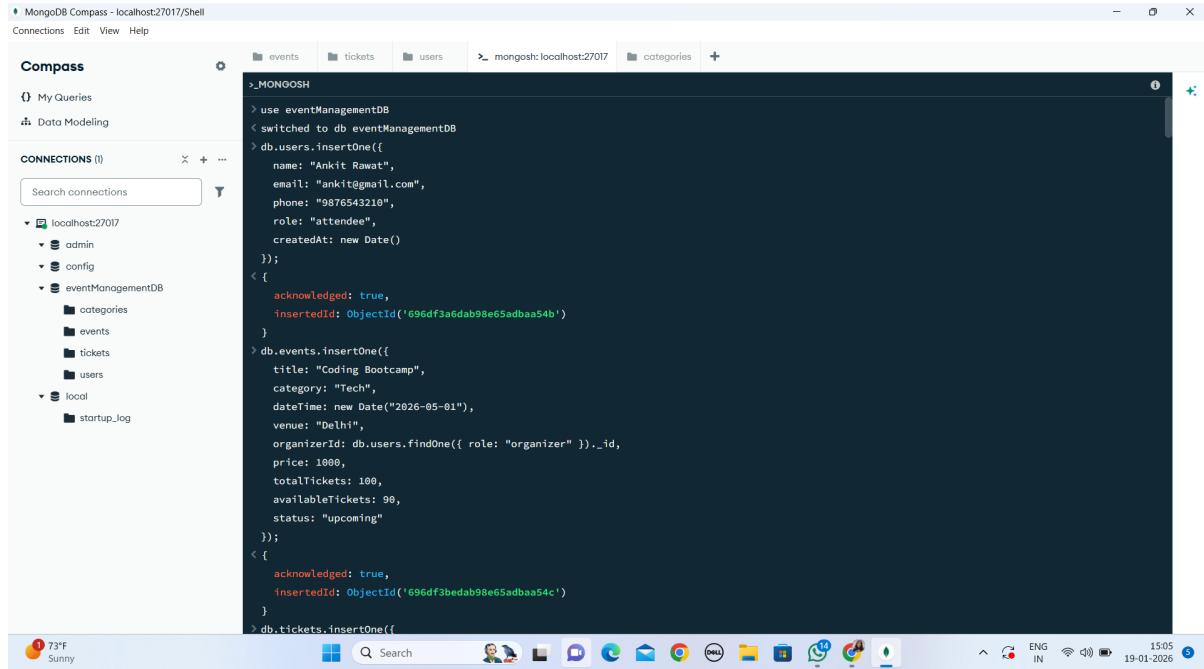
_id: ObjectId("696df2663a0c9b20ab383100")
title: "Rock Music Night"
description: "Live rock concert"
```

This screenshot shows the MongoDB Compass interface for the 'events' collection, displaying 22 documents. The interface is similar to the previous one, with a sidebar and a main document list area. The table structure for the documents is as follows:

	_id	ObjectId	title	String	description	String	categoryId	ObjectId	dateTime	Date
7	ObjectId	"696df2663a0c9b2...	"Art Expo"		"Art & craft exhibition"	No field			2026-04-01T00:00:00.000+...	
8	ObjectId	"696df2663a0c9b2...	"Photography Workshop"		"Basics of photography"	No field			2026-04-10T00:00:00.000+...	
9	ObjectId	"696df2663a0c9b2...	"Football Tournament"		"Inter-college football"	No field			2026-03-01T00:00:00.000+...	
10	ObjectId	"696df2663a0c9b2...	"Cricket League"		"Local cricket matches"	No field			2026-02-18T00:00:00.000+...	
11	ObjectId	"696df2663a0c9b2...	"Yoga Retreat"		"Wellness & yoga"	No field			2026-03-15T00:00:00.000+...	
12	ObjectId	"696df2663a0c9b2...	"Meditation Camp"		"Mindfulness training"	No field			2026-03-28T00:00:00.000+...	
13	ObjectId	"696df2663a0c9b2...	"Rock Music Night"		"Live rock concert"	No field			2026-02-01T00:00:00.000+...	
14	ObjectId	"696df2663a0c9b2...	"Jazz Evening"		"Smooth jazz performance"	No field			2026-02-05T00:00:00.000+...	
15	ObjectId	"696df2663a0c9b2...	"Tech Summit 2026"		"AI & Cloud Conference"	No field			2026-03-10T00:00:00.000+...	
16	ObjectId	"696df2663a0c9b2...	"Startup Meetup"		"Entrepreneur networking"	No field			2026-01-20T00:00:00.000+...	
17	ObjectId	"696df2663a0c9b2...	"Art Expo"		"Art & craft exhibition"	No field			2026-04-01T00:00:00.000+...	
18	ObjectId	"696df2663a0c9b2...	"Photography Workshop"		"Basics of photography"	No field			2026-04-10T00:00:00.000+...	
19	ObjectId	"696df2663a0c9b2...	"Football Tournament"		"Inter-college football"	No field			2026-03-01T00:00:00.000+...	
20	ObjectId	"696df2663a0c9b2...	"Cricket League"		"Local cricket matches"	No field			2026-02-18T00:00:00.000+...	
21	ObjectId	"696df2663a0c9b2...	"Yoga Retreat"		"Wellness & yoga"	No field			2026-03-15T00:00:00.000+...	
22	ObjectId	"696df2663a0c9b2...	"Meditation Camp"		"Mindfulness training"	No field			2026-03-28T00:00:00.000+...	

5. Tickets Collection

The Tickets collection stores booking information. Users can book multiple tickets for an event.



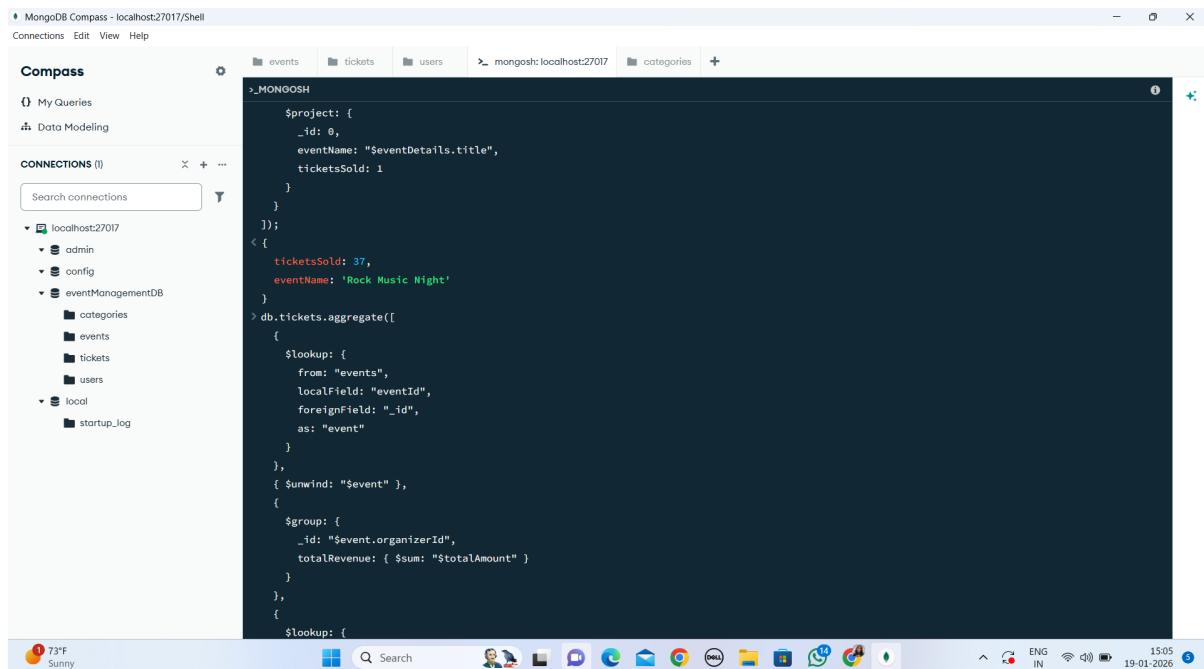
The screenshot shows the MongoDB Compass interface connected to the 'eventManagementDB' database. The left sidebar displays the database structure with collections: categories, events, tickets, users, and startup_log. The 'tickets' collection is currently selected. The main pane shows the MongoDB shell with the following command history:

```
> use eventManagementDB
< switched to db eventManagementDB
> db.users.insertOne({
  name: "Ankit Rawat",
  email: "ankit@gmail.com",
  phone: "9876543210",
  role: "attendee",
  createdAt: new Date()
});
< [
  {
    acknowledged: true,
    insertedId: ObjectId('696df3a6dab98e65adbaa54b')
  }
]
> db.events.insertOne({
  title: "Coding Bootcamp",
  category: "Tech",
  date: new Date("2026-05-01"),
  venue: "Delhi",
  organizerId: db.users.findOne({ role: "organizer" })._id,
  price: 1000,
  totalTickets: 100,
  availableTickets: 99,
  status: "upcoming"
});
< [
  {
    acknowledged: true,
    insertedId: ObjectId('696df3bedab98e65adbaa54c')
  }
]
> db.tickets.insertOne({
```

The status bar at the bottom indicates the system is 'Sunny' at 73°F, the date is 19-01-2026, and the time is 15:05.

6. CRUD Operations

CRUD operations were performed on all collections including inserting new data, updating records, fetching data, and deleting records.

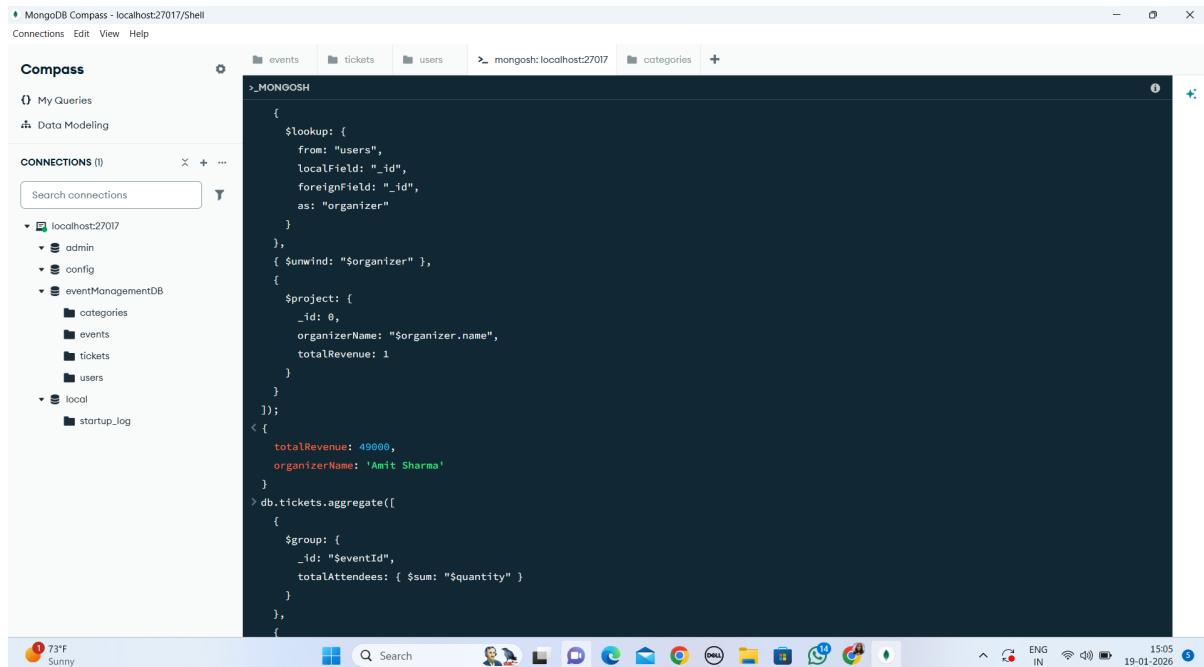


The screenshot shows the MongoDB Compass interface connected to the 'localhost:27017' shell. The left sidebar displays the 'Compass' menu, 'My Queries', 'Data Modeling', and a 'CONNECTIONS' section listing 'localhost:27017' (admin, config, eventManagementDB, local, startup_log) and 'mongosh: localhost:27017' (events, tickets, users). The main panel shows an aggregation pipeline starting with a '\$group' stage on the 'MONGOSH' collection, followed by a '\$lookup' stage to join with the 'events' collection, and finally a '\$group' stage to calculate total revenue per organizer. The bottom taskbar shows system icons and the date/time: 19-01-2026, 15:05.

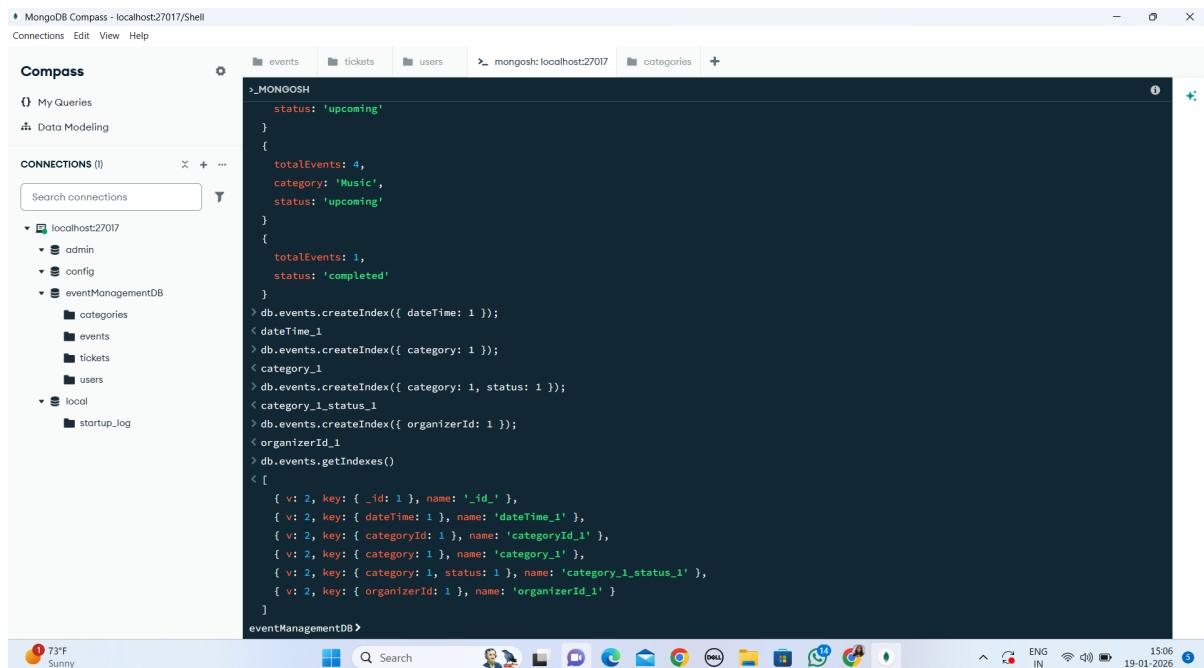
```
>_MONGOSH
{
  $group: {
    _id: 0,
    eventName: "$eventDetails.title",
    ticketsSold: 1
  }
}
]);
< {
  ticketsSold: 37,
  eventName: 'Rock Music Night'
}
> db.tickets.aggregate([
  {
    $lookup: {
      from: "events",
      localField: "eventId",
      foreignField: "_id",
      as: "event"
    }
  },
  { $unwind: "$event" },
  {
    $group: {
      _id: "event.organizerId",
      totalRevenue: { $sum: "$totalAmount" }
    }
  },
  {
    $lookup: {
```

7. Aggregation Pipelines

Aggregation pipelines were used to generate reports such as top events by ticket sales, total revenue per organizer, and attendees per event.



```
>_MONGOSH
{
  $lookup: {
    from: "users",
    localField: "_id",
    foreignField: "_id",
    as: "organizer"
  },
  { $unwind: "$organizer" },
  {
    $project: {
      _id: 0,
      organizerName: "$organizer.name",
      totalRevenue: 1
    }
  }
);
< {
  totalRevenue: 49000,
  organizerName: 'Amit Sharma'
}
> db.tickets.aggregate([
  {
    $group: {
      _id: "$eventId",
      totalAttendees: { $sum: "$quantity" }
    }
  },
  {
    $sort: {
      totalAttendees: -1
    }
  }
]);
< [
  { _id: "e1", totalAttendees: 10000 },
  { _id: "e2", totalAttendees: 9000 },
  { _id: "e3", totalAttendees: 8000 },
  { _id: "e4", totalAttendees: 7000 },
  { _id: "e5", totalAttendees: 6000 }
]
```



```
>_MONGOSH
{
  status: 'upcoming'
}
{
  totalEvents: 4,
  category: 'Music',
  status: 'upcoming'
}
{
  totalEvents: 1,
  status: 'completed'
}
> db.events.createIndex({ dateTime: 1 });
< dateTime_1
> db.events.createIndex({ category: 1 });
< category_1
> db.events.createIndex({ category: 1, status: 1 });
< category_1_status_1
> db.events.createIndex({ organizerId: 1 });
< organizerId_1
> db.events.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_1' },
  { v: 2, key: { dateTime: 1 }, name: 'dateTime_1' },
  { v: 2, key: { categoryId: 1 }, name: 'categoryId_1' },
  { v: 2, key: { category: 1 }, name: 'category_1' },
  { v: 2, key: { category: 1, status: 1 }, name: 'category_1_status_1' },
  { v: 2, key: { organizerId: 1 }, name: 'organizerId_1' }
]
eventManagementDB
```

MongoDB Compass - localhost:27017/eventManagementDB

Connections Edit View Help

Compass

My Queries Data Modeling

CONNECTIONS (1)

- Search connections
- localhost:27017
 - admin
 - config
 - eventManagementDB**
 - categories
 - events
 - tickets
 - users
 - local
 - startup_log

localhost:27017 > eventManagementDB

Open MongoDB shell Create collection Refresh

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
categories	-	36.86 kB	20	72.00 B	1	36.86 kB
events	-	36.86 kB	22	247.00 B	3	110.59 kB
tickets	-	36.86 kB	20	134.00 B	2	73.73 kB
users	-	36.86 kB	47	132.00 B	1	36.86 kB

72°F Sunny Search ENG IN 15:06 19-01-2026

8. Indexes

Indexes were created on frequently searched fields such as event date and category to improve query performance.

9. Conclusion

This assignment demonstrates how MongoDB can be effectively used for a real-world event management application. The schema design, CRUD operations, and aggregations fulfill all assignment requirements.