Full Stack Development with MERN

Database Design and Development Report

Date: 08/07/2024

Team ID: SWTID1743520385

Project Name: Book a Doctor Platform

Maximum Marks: —

# Project Title: Book a Doctor Platform

Date: 08/07/2024

Prepared by: SWTID1743520385

# Objective

The objective of this report is to outline the database design and implementation details for the Book a Doctor project, including schema modeling and database integration using MongoDB Atlas.

# Technologies Used

• Database Management System (DBMS): MongoDB Atlas  
• Object-Document Mapper (ODM): Mongoose

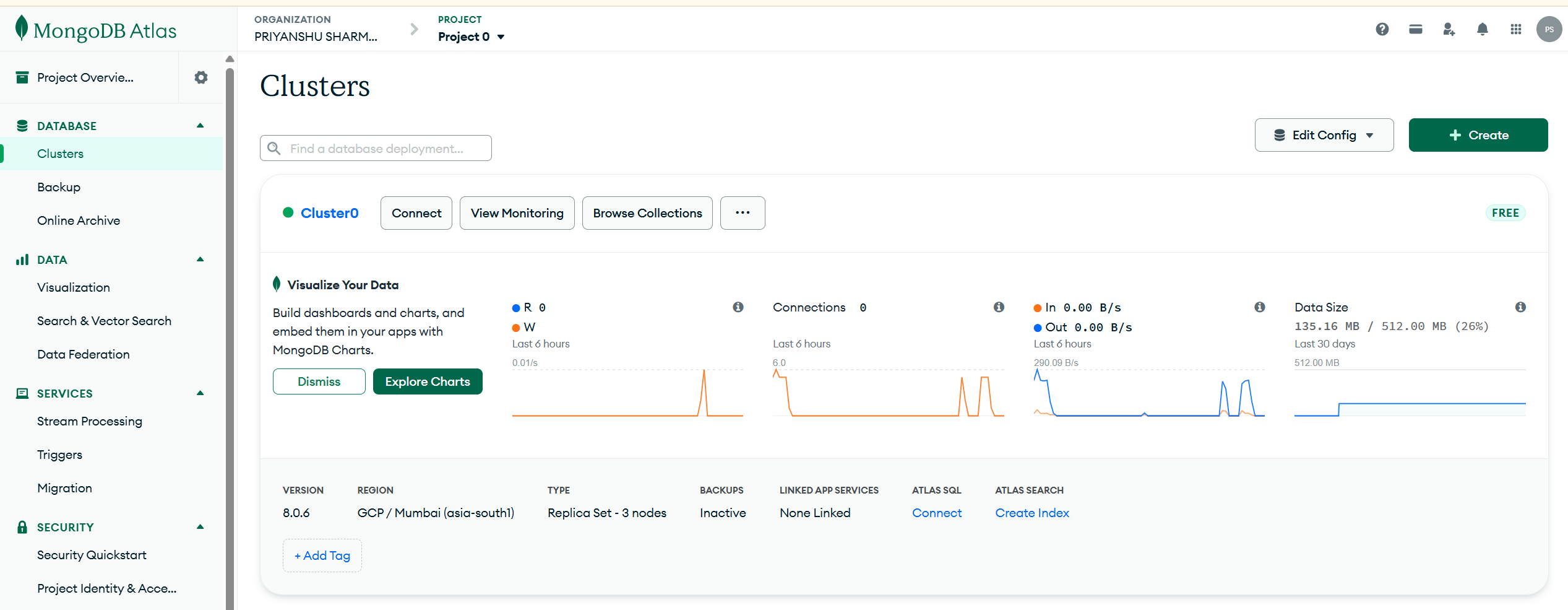
# Design the Database Schema

The database schema is designed to manage core entities including users (patients/doctors), appointments, and admin data.

1. Users (Patients & Doctors)  
 - Attributes:  
• name: { type: String, required: true }  
• email: { type: String, required: true, unique: true }  
• password: { type: String, required: true }  
• role: { type: String, enum: ['patient', 'doctor', 'admin'] }

2. Appointments  
 - Attributes:  
• patientId: ObjectId (ref: User)  
• doctorId: ObjectId (ref: User)  
• appointmentDate: Date  
• timeSlot: String  
• status: { type: String, enum: ['scheduled', 'completed', 'cancelled'] }

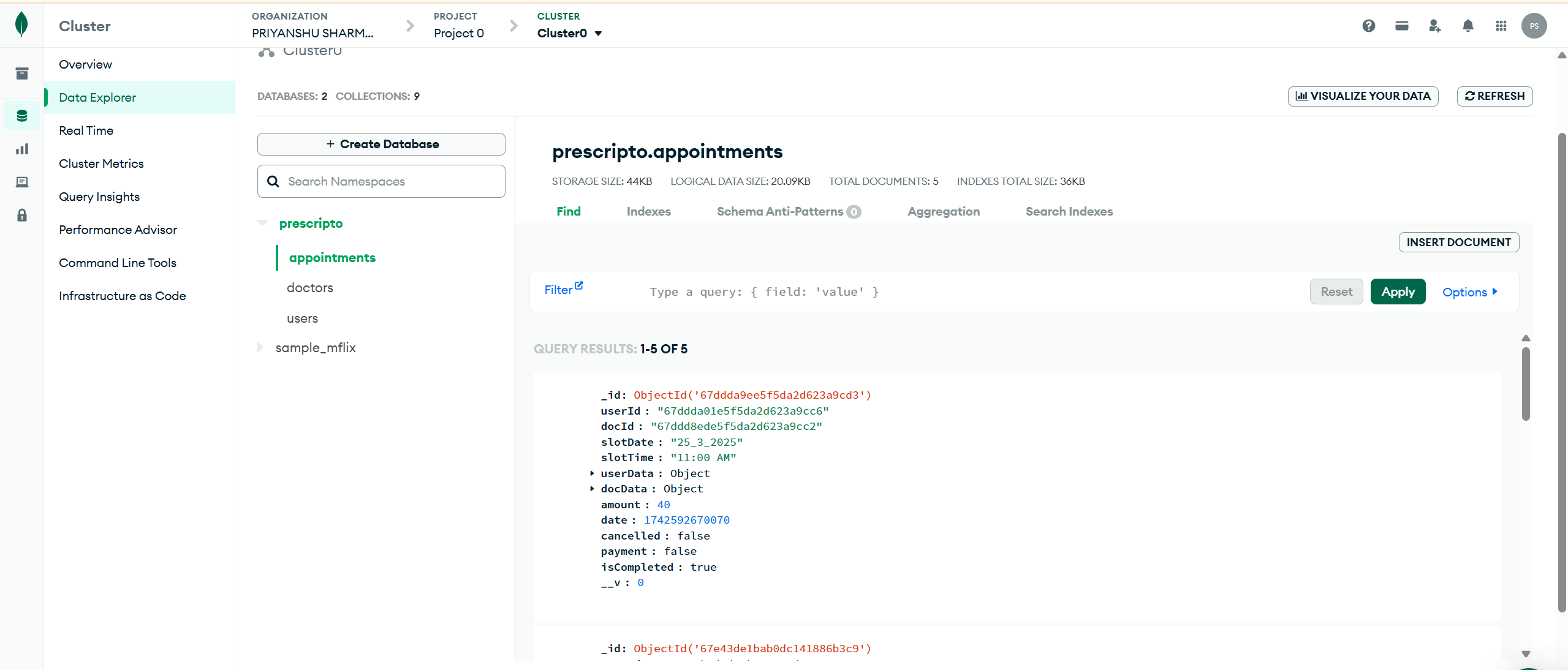
# Implemented Collections using MongoDB Atlas



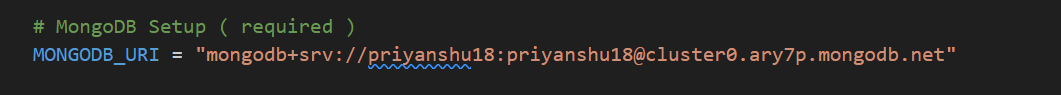
Database Name: book-a-doctor

1. Collection: users  
 - Schema:  
{  
 \_id: ObjectId,  
 name: String,  
 email: String,  
 password: String,  
 role: String,  
 createdAt: Date,  
 updatedAt: Date  
}

2. Collection: appointments  
 - Schema:  
{  
 \_id: ObjectId,  
 patientId: ObjectId,  
 doctorId: ObjectId,  
 appointmentDate: Date,  
 timeSlot: String,  
 status: String,  
 createdAt: Date,  
 updatedAt: Date  
}



# Integration with Backend



The backend communicates with MongoDB Atlas using Mongoose. Key operations include CRUD operations for users and appointments.

• User Registration Example:  
const user = await User.findOne({ email: req.body.email });  
if (!user) {  
 const newUser = new User(req.body);  
 await newUser.save();  
 res.status(201).json({ success: 'User created successfully' });  
} else {  
 res.status(409).json({ success: 'Email already exists' });  
}

• Appointment Booking Example:  
const newAppointment = new Appointment(req.body);  
await newAppointment.save();  
res.status(201).json({ success: 'Appointment booked successfully' });