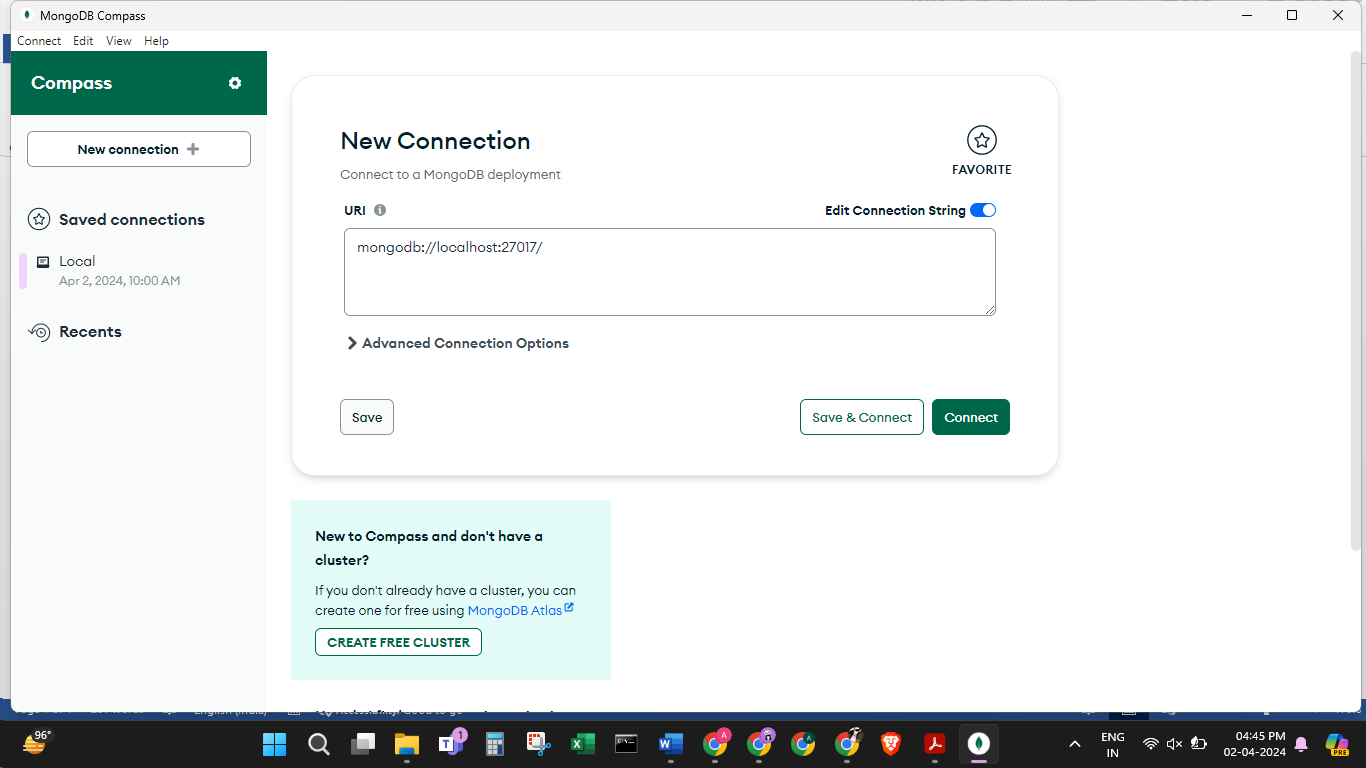
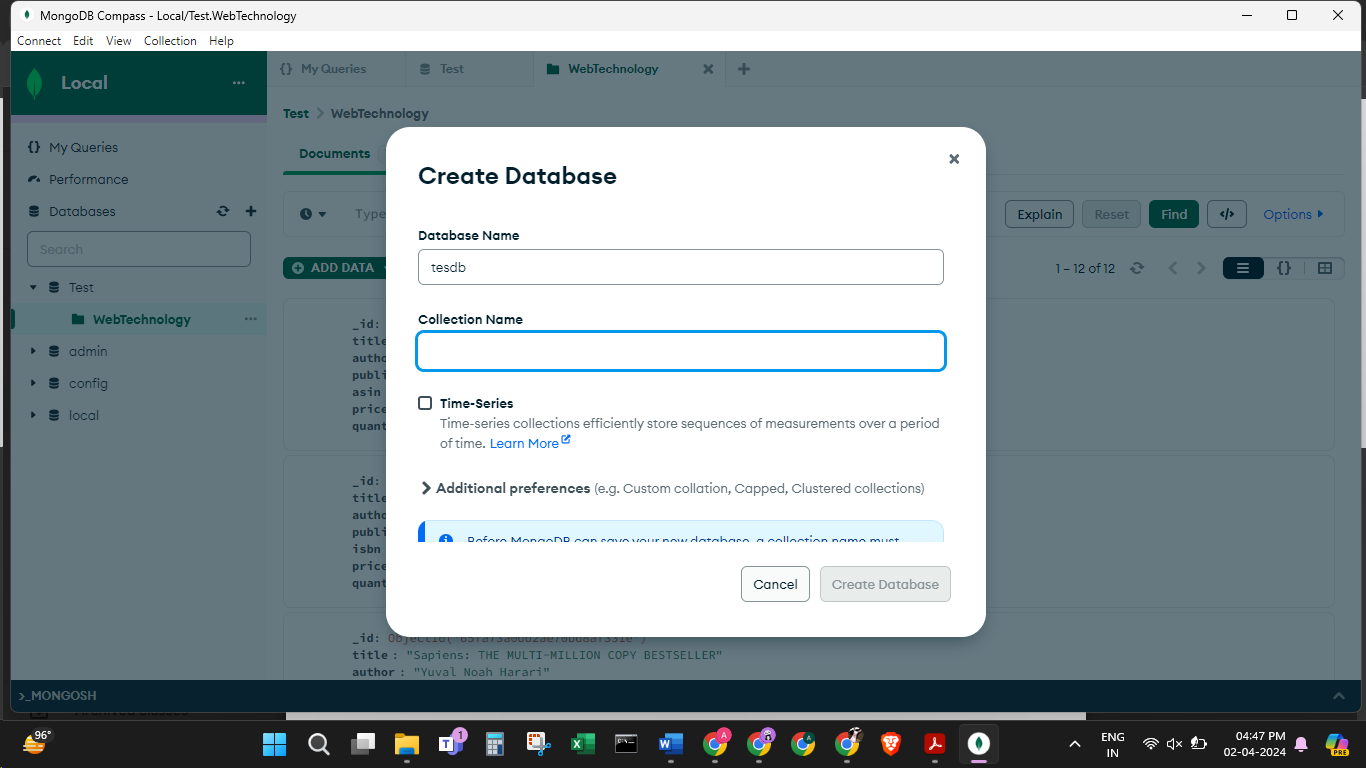
**NAME: SHANGANA YADAV**

**ROLL NO: 22IT3045**

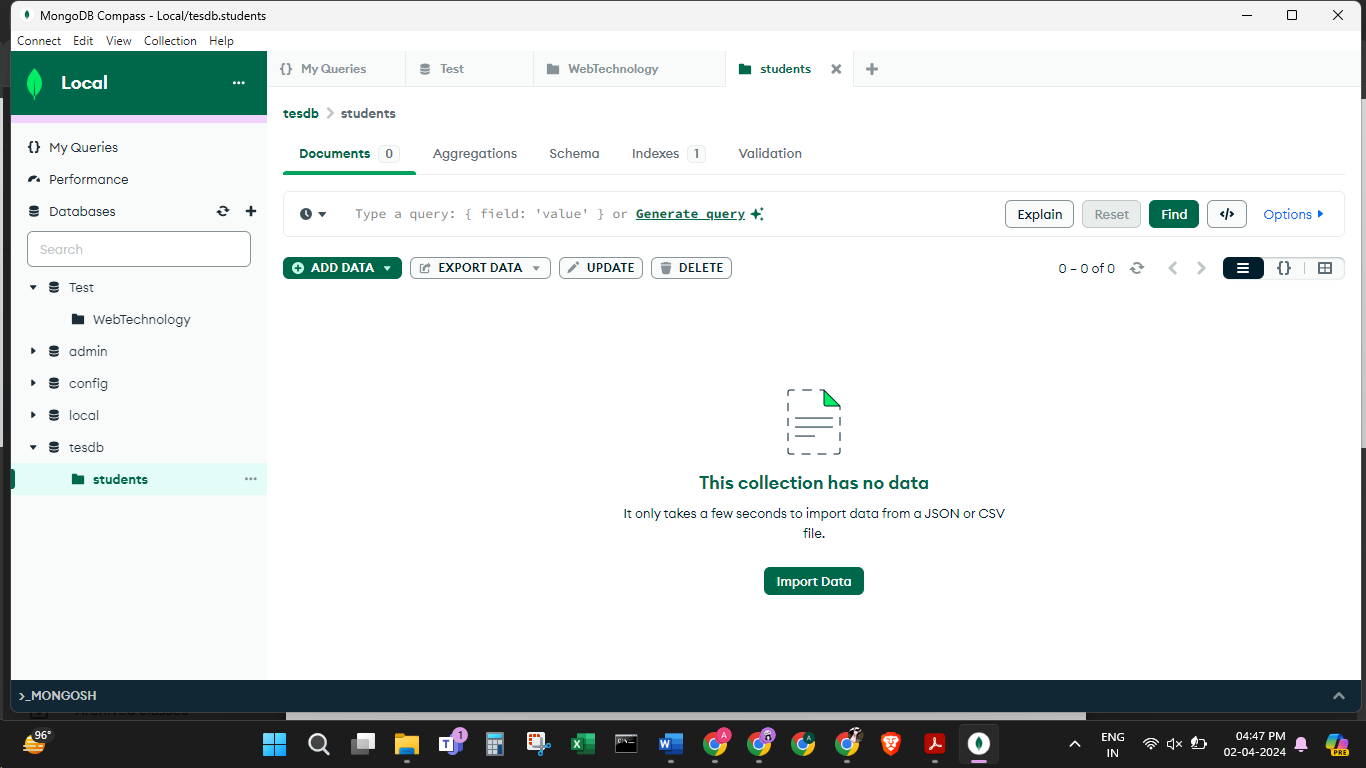
1. Connect to a MongoDB server using MongoDB Compass.



2. Create a new database named "testdb" in MongoDB Compass.



3.Create a new collection named "students" in the "testdb" database.



4. Insert ten documents into the "students" collection with the following fields:

name, age, and email.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

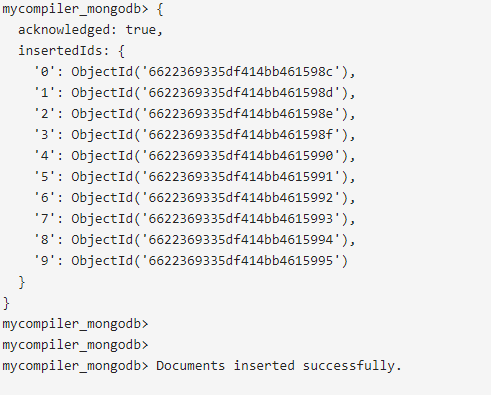
];

// Insert documents into the "students" collection

db.students.insertMany(studentsData);

// Confirm insertion

print("Documents inserted successfully.");



5.View the contents of the "students" collection.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

];

// Insert documents into the "students" collection

db.students.insertMany(studentsData);

// Confirm insertion

print("Documents inserted successfully.");

db.students.find()





6.Update the age of a specific student in the "students" collection.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

];

// Insert documents into the "students" collection

db.students.insertMany(studentsData);

// Confirm insertion

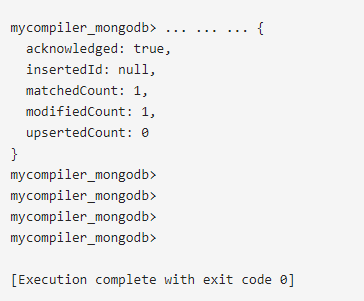
print("Documents inserted successfully.");

db.students.updateOne(

{ name: "Ravija Chandel" },

{ $set: { age: 21 } }

)



7. Delete a document from the "students" collection based on a specific

condition.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

];

// Insert documents into the "students" collection

db.students.insertMany(studentsData);

// Confirm insertion

print("Documents inserted successfully.");

db.students.deleteOne({ name: "Aayona Somvanshi" })



8.Use the aggregation pipeline to calculate the average age of all students in the

"students" collection.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

];

// Insert documents into the "students" collection

db.students.insertMany(studentsData);

// Confirm insertion

print("Documents inserted successfully.");

db.students.aggregate([

{

$group: {

\_id: null,

averageAge: { $avg: "$age" }

}

}

])



9.Create an index on the "name" field in the "students" collection.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

];

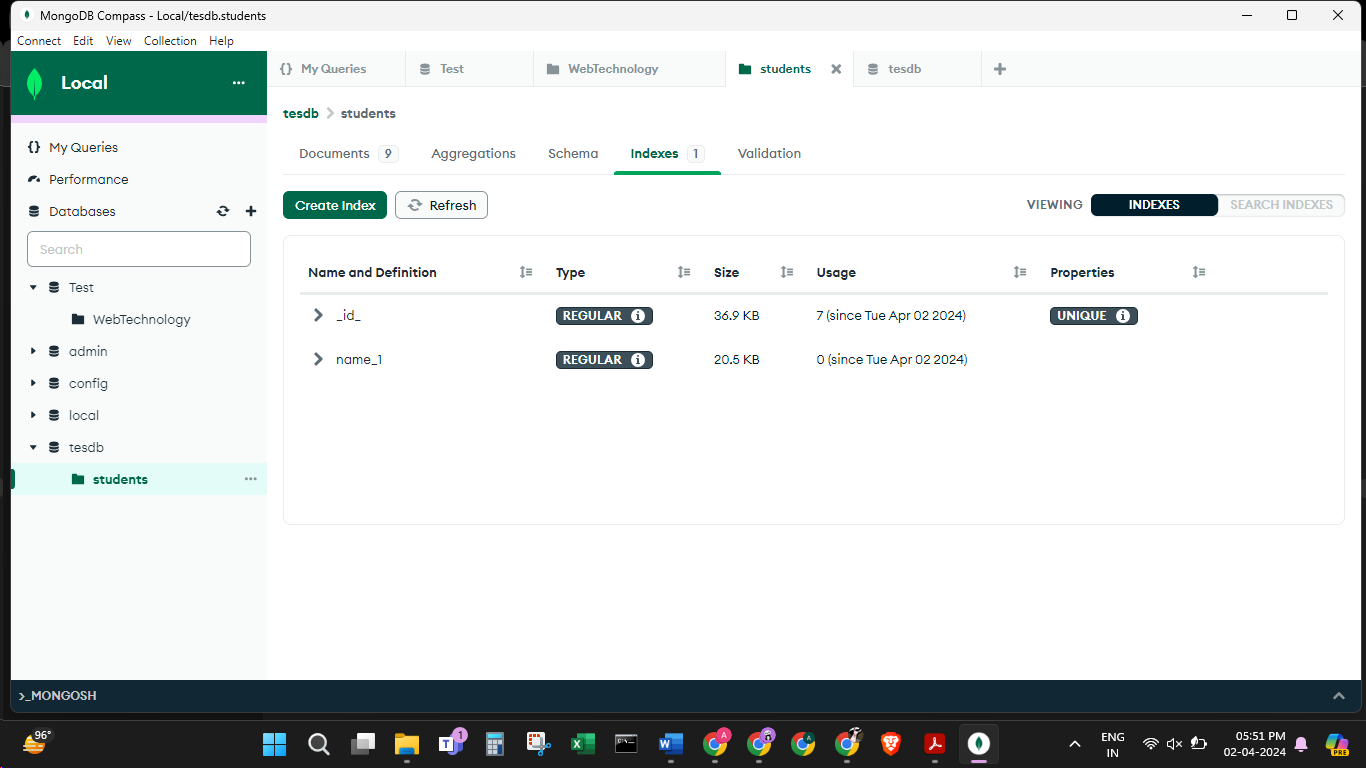
// Insert documents into the "students" collection

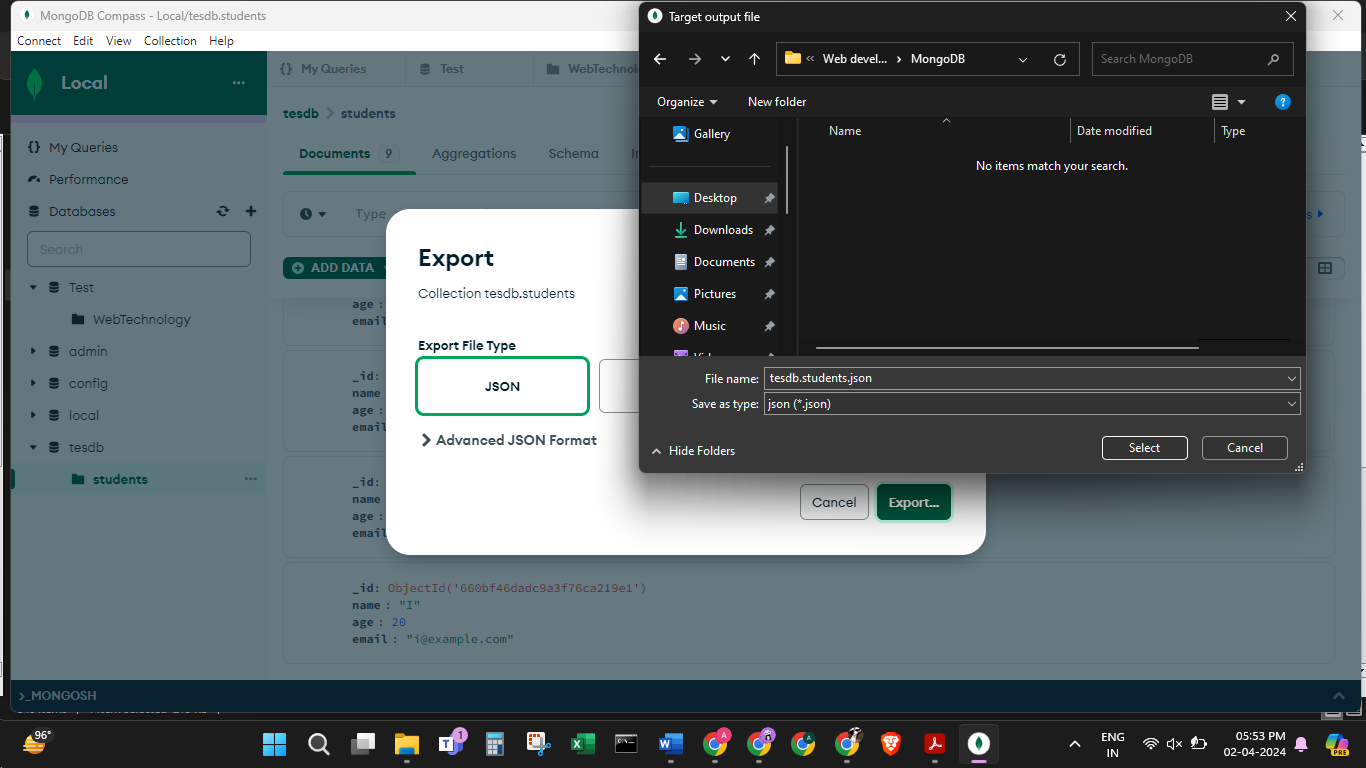
db.students.insertMany(studentsData);

// Confirm insertion

print("Documents inserted successfully.");

db.students.createIndex({ name: 1 })

10. Export the contents of the "students" collection to a JSON file.



11.Perform a complex aggregation operation to find the top 5 oldest students in

the "students" collection.

var studentsData = [

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

{ name: "Aayona Somvanshi", age: 21, email: "aayona.somvanshi@example.com" },

{ name: "Ravija Chandel", age: 20, email: "ravija.chandel@example.com" },

{ name: "Shangana Yadav", age: 21, email: "shangana.yadav@example.com" },

];

// Insert documents into the "students" collection

db.students.insertMany(studentsData);

// Confirm insertion

print("Documents inserted successfully.");

db.students.aggregate([

// Sort documents by age in descending order

{ $sort: { age: -1 } },

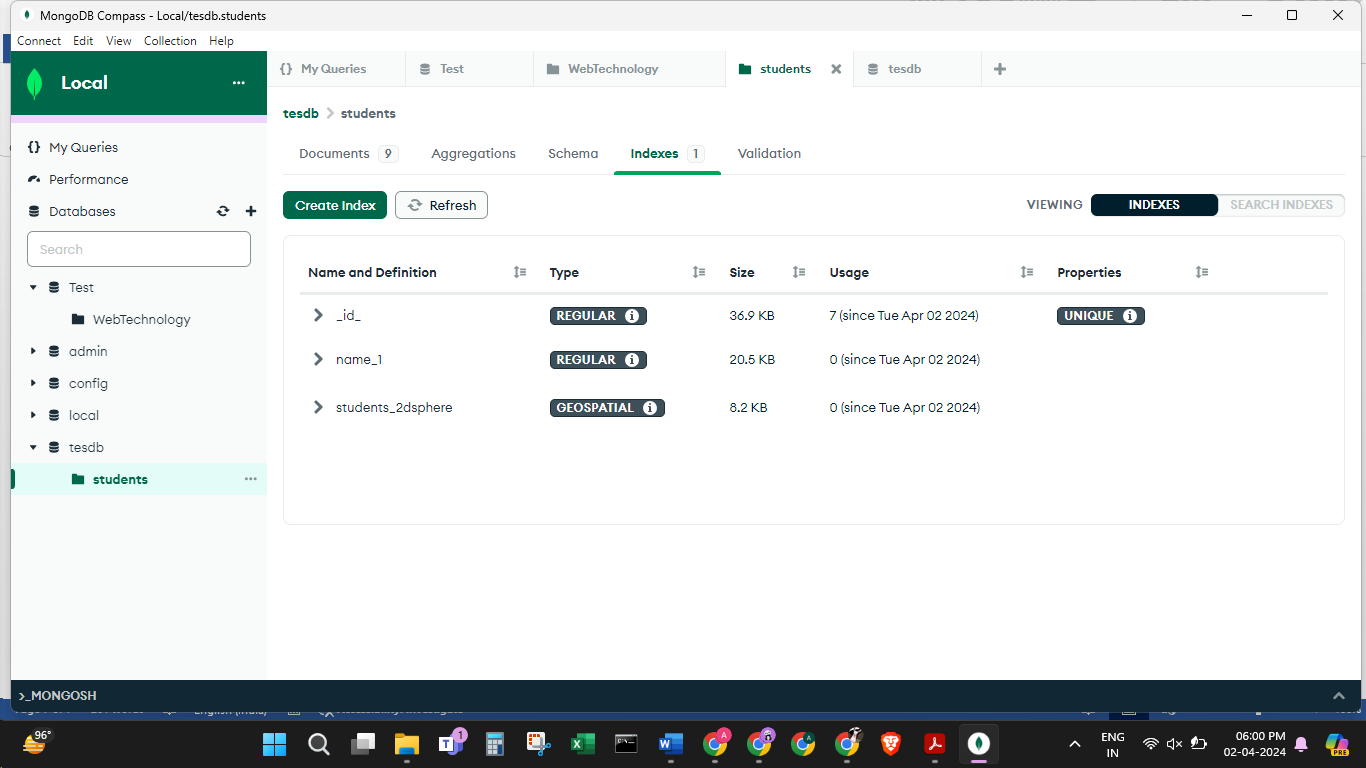
// Limit to the top 5 documents

{ $limit: 5 }

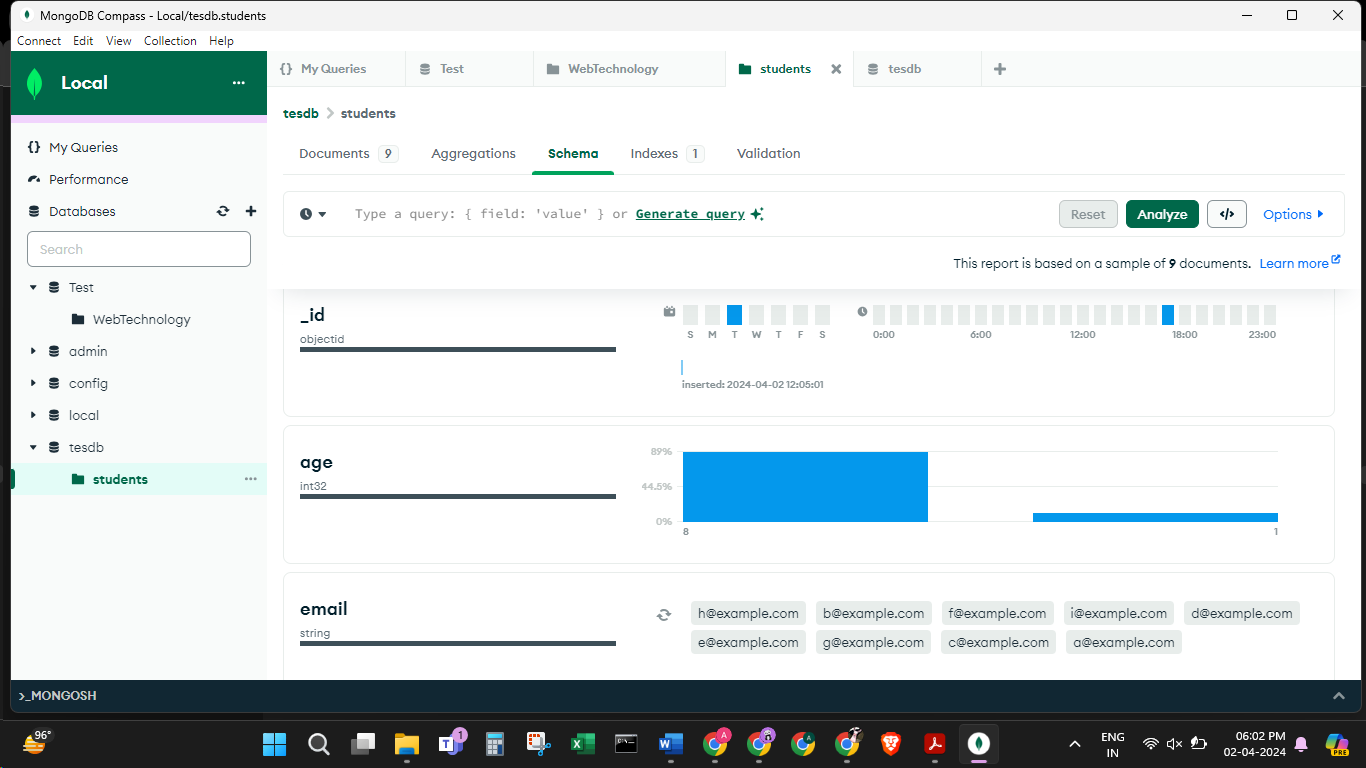
])



12. Create a geospatial index on a field representing the location of students.

13.Use MongoDB Compass to visualize the data distribution in the "students"

collection.



14. Set up a data validation rule to ensure that documents in the "students"

collection must have a non-empty name field.

