**Course Code:** CSE 3513

**Course Name:** NoSQL Data Management

**Course Instructor**: Sunil Sahoo

**Lab Session**: 01

**Activity**: Setting Up Environment for MongoDB

# Learning Objectives (Los)

**LO1:** Understand the system prerequisites for installing MongoDB.

**LO2:** Install MongoDB Community Edition on Windows/Linux/macOS.

**LO3:** Set up MongoDB service and validate installation.

**LO4:** Use MongoDB Shell (mongosh) and Compass GUI for basic operations.

**LO5:** Perform basic CRUD operations using MongoDB shell.

MongoDB is a popular **NoSQL document database** designed for high availability, scalability, and flexibility. Unlike traditional relational databases, MongoDB stores data in **JSON-like documents** (BSON), making it ideal for applications that deal with large volumes of semi-structured or unstructured data.

Before diving into how MongoDB works internally or exploring real-world use cases like **e-commerce**, **event logging**, or **recommendation engines**, it is crucial to set up the local development environment. This activity ensures that learners can install MongoDB, connect to it using both the **command-line shell (mongosh)** and the **GUI (Compass)**, and perform basic database operations confidently.

This setup will act as the **foundation for all subsequent hands-on labs and projects** in the course.

**Summary:** This activity introduces participants to setting up a MongoDB environment, both through the command-line interface and GUI tools. By the end of the session, learners will have a functional MongoDB instance on their local machines, understand its architecture, and be able to perform basic database operations. This foundational setup is critical before exploring real-world use cases and case studies like content management systems and e-commerce applications using MongoDB.

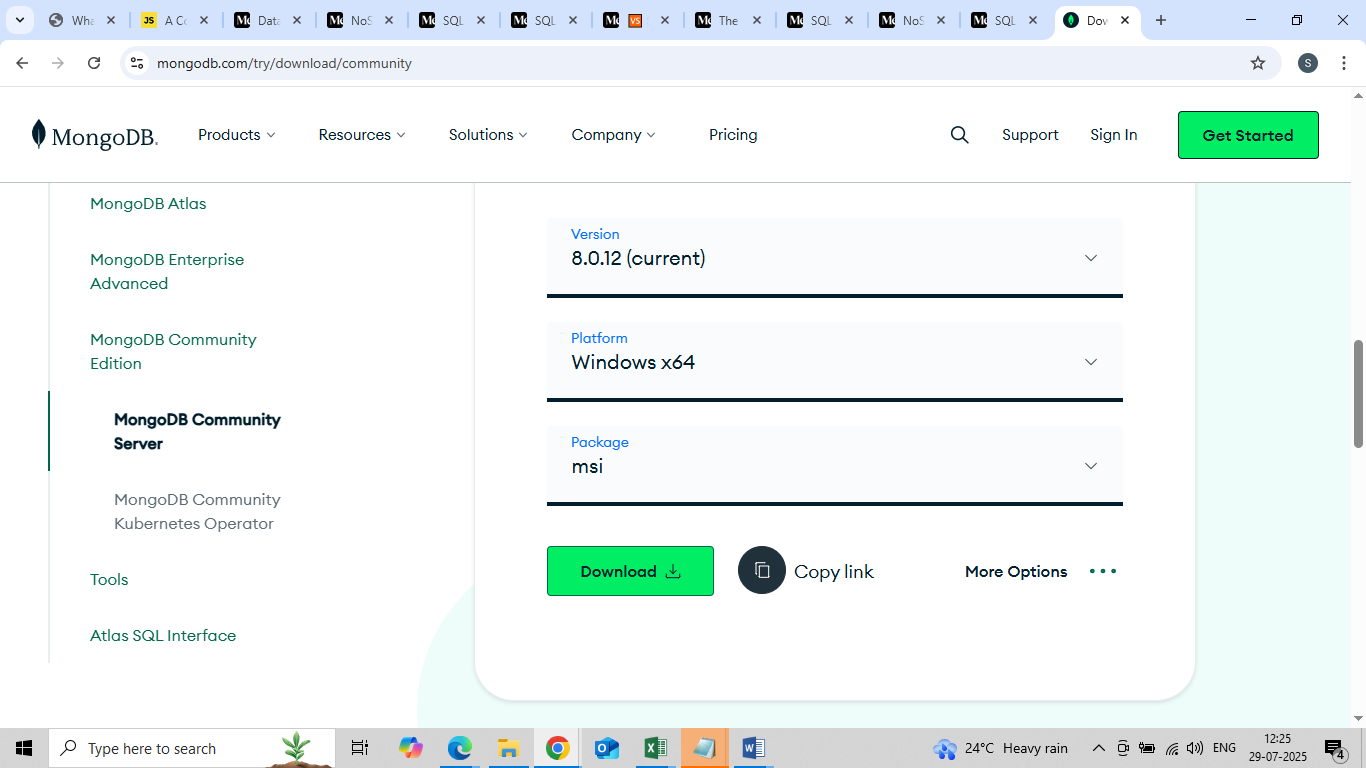
**Activity Steps**

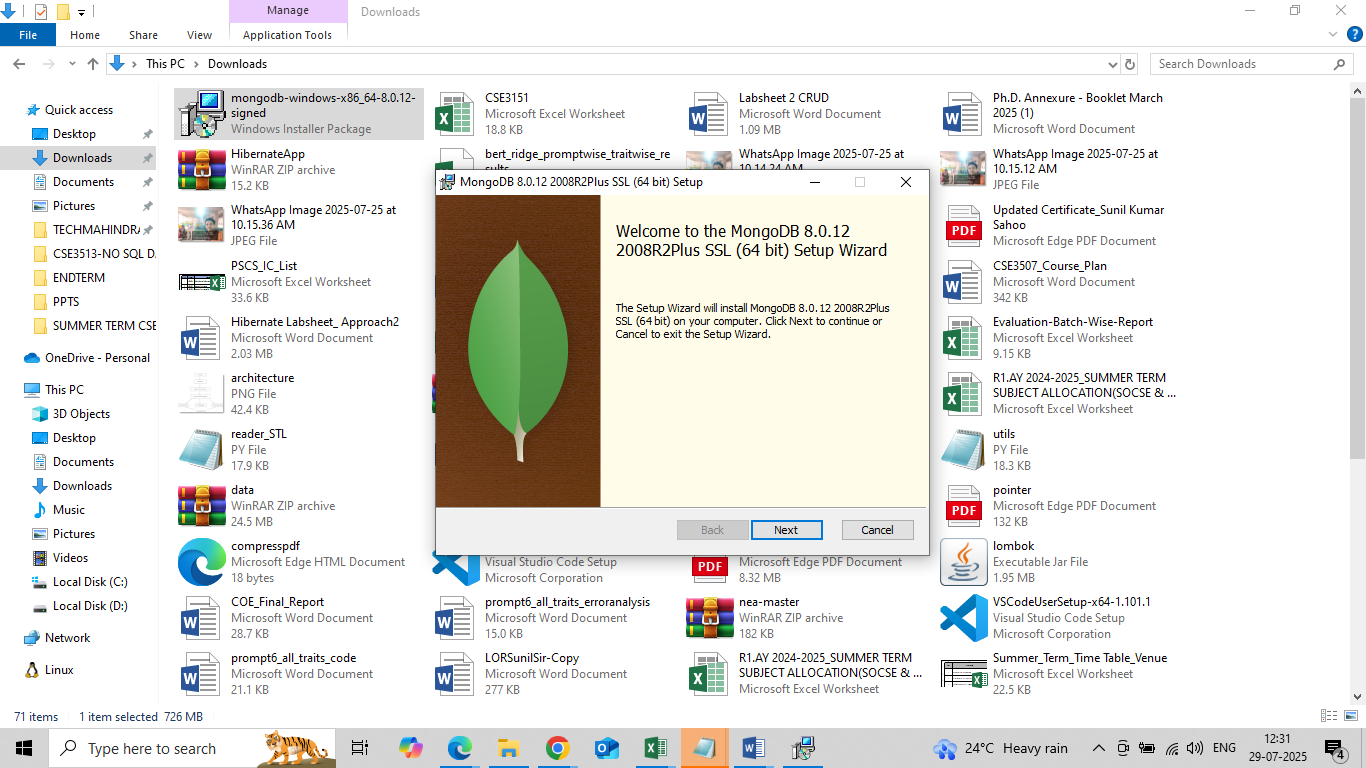
**Step 1:** Check System requirements

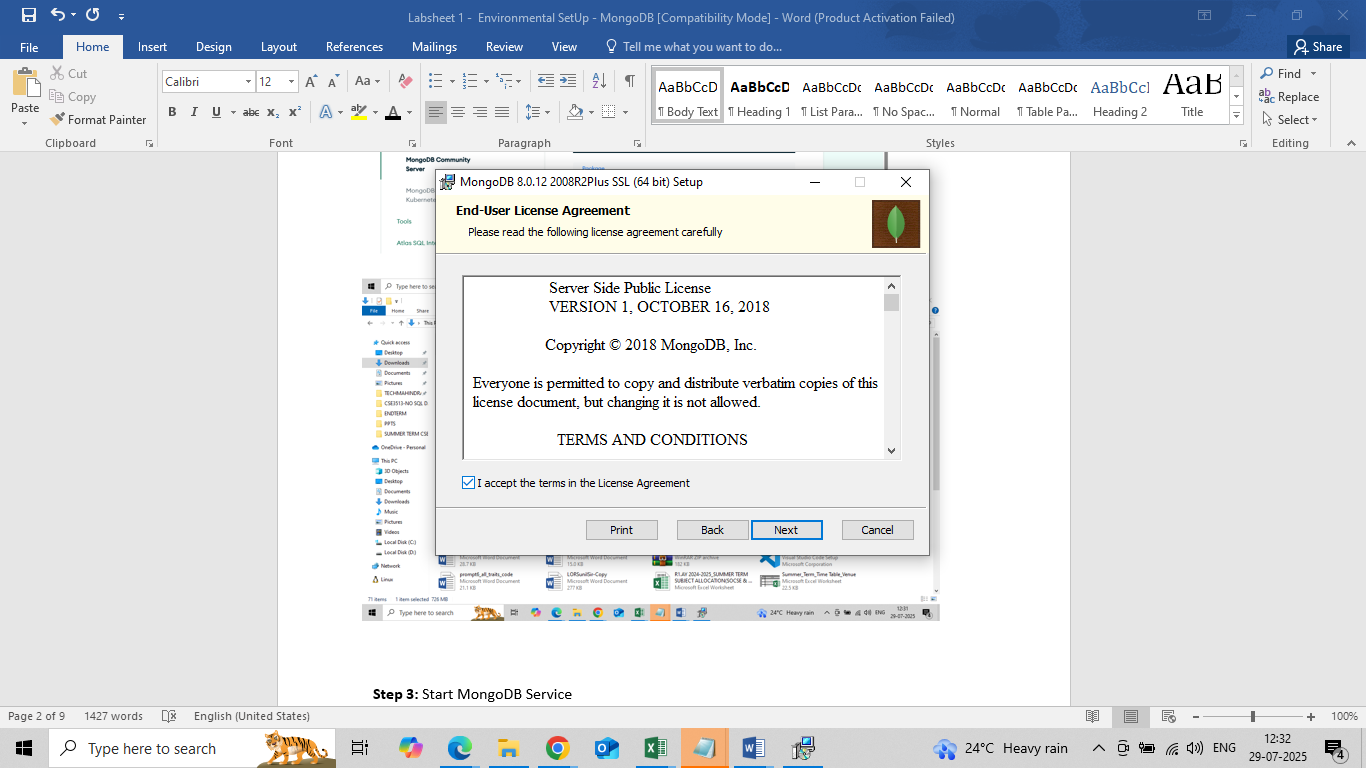
* + OS Version : Windows 10 or later, 64bits
  + Memory disk space : minimum 2 GB RAM, Minimum 5 GB Free Disk space
  + Internet Connection : Required for downloading packages or docker images (optional)
  + Admin Privileges : Required

**Step 2:** Download and Install MongoDB

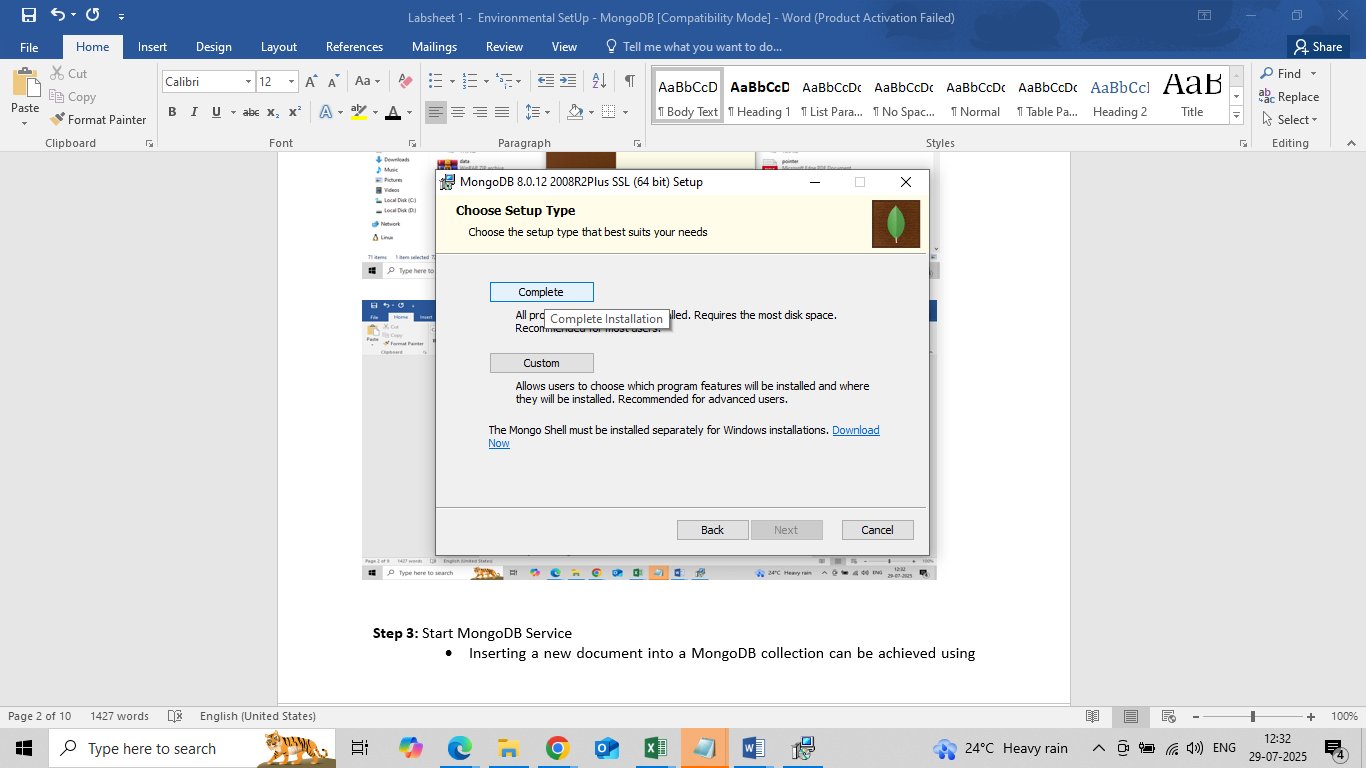
* + Visit <https://www.mongodb.com/try/download/community>
  + Choose your OS (Windows/macOS/Linux).
  + Follow installation guide based on the OS.

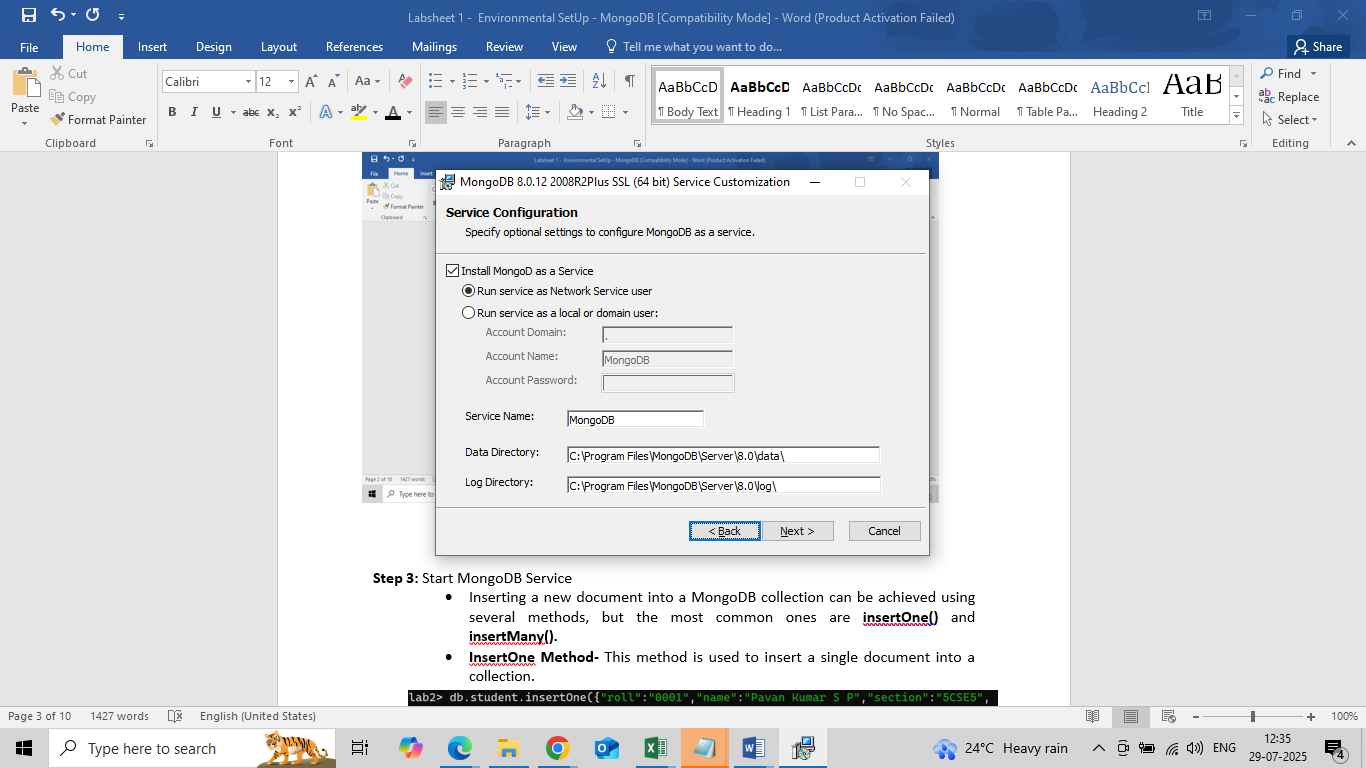


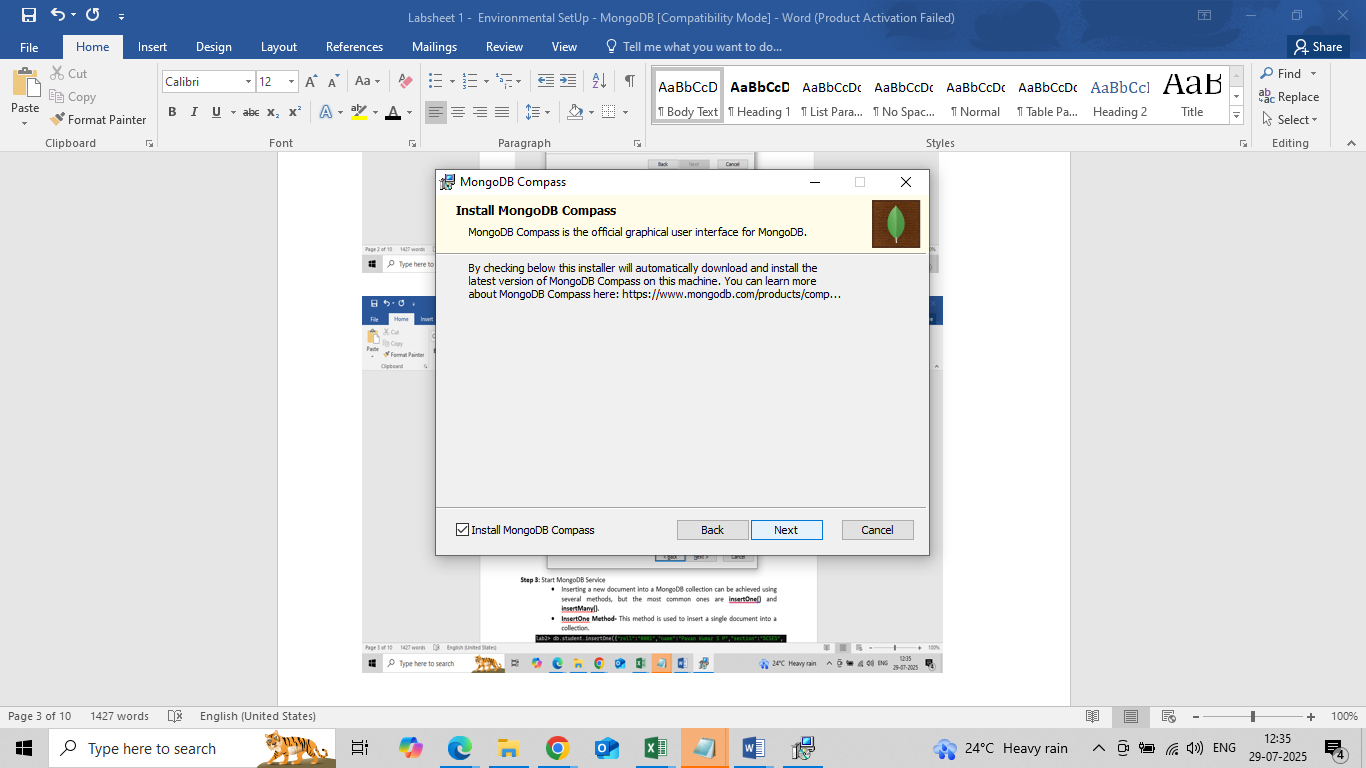


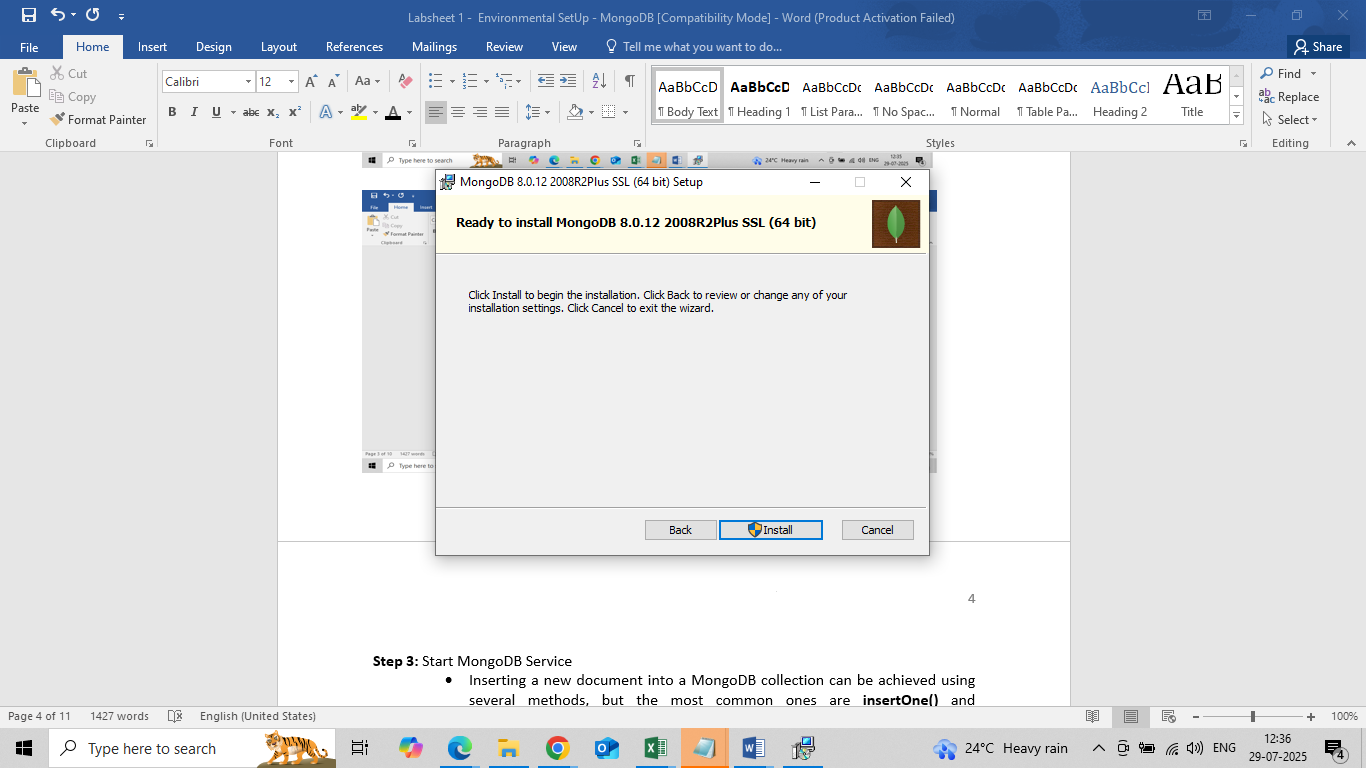


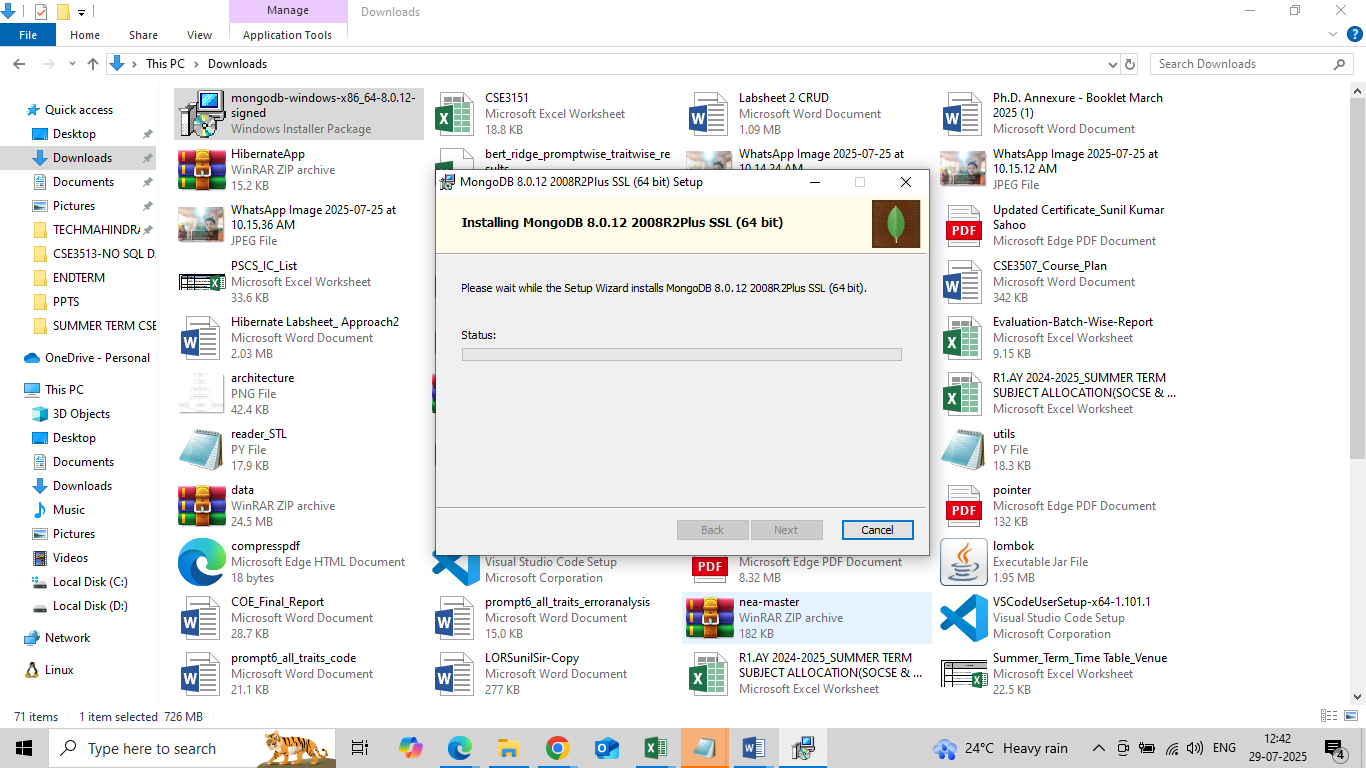
Choose Complete for C Drive , Custom for other Drive, Usually if C Drive has not enough space (5GB min)



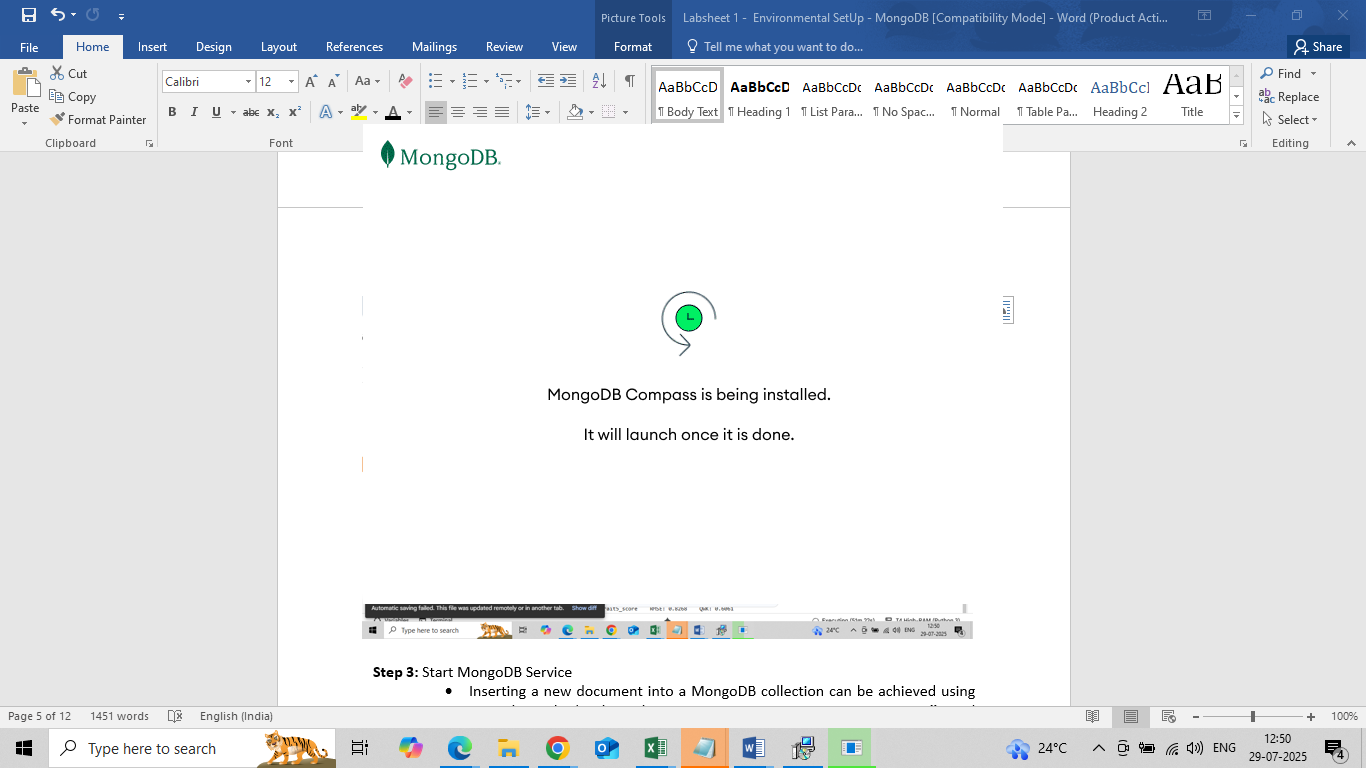


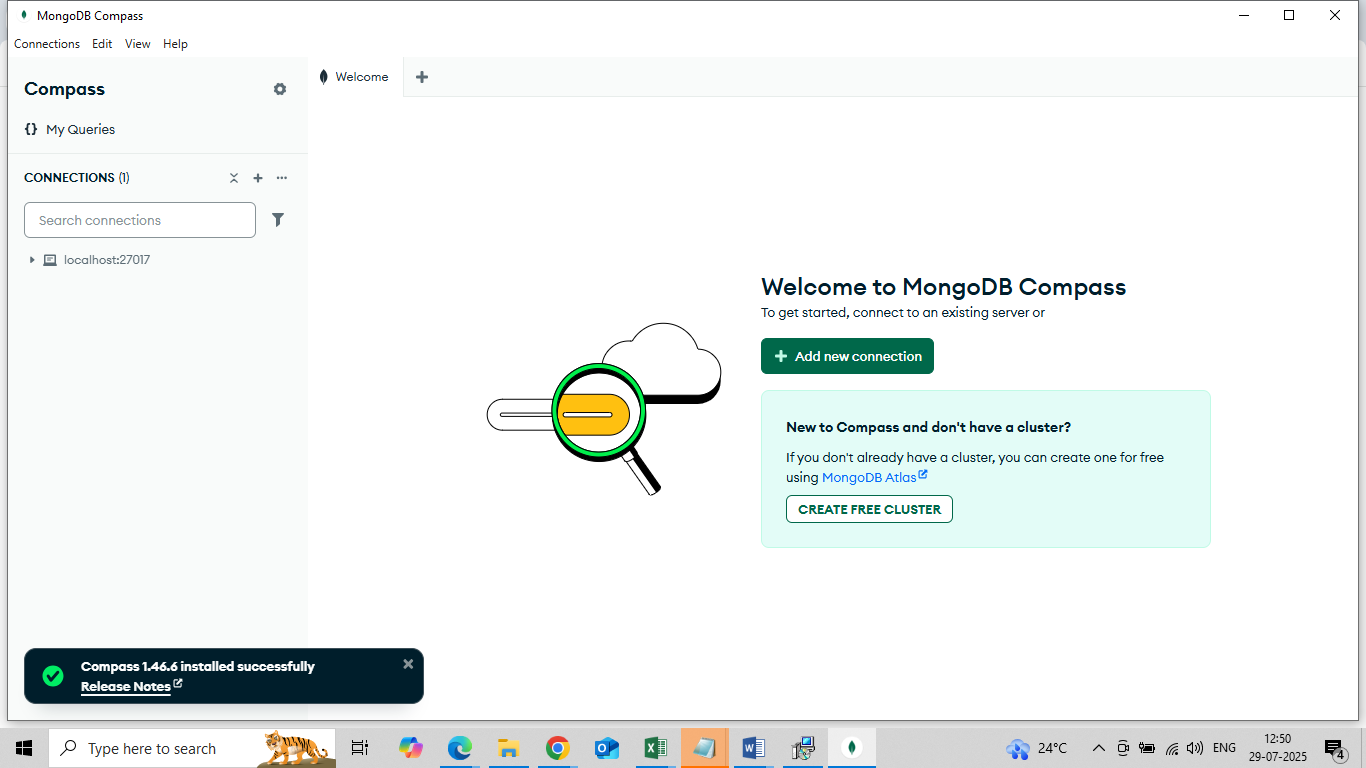






5mins-10mins (depends of system)





Compass 1.46.6 installed successfully.

MongoDB Compass/client is a GUI tool to connect to and work with your MongoDB database.

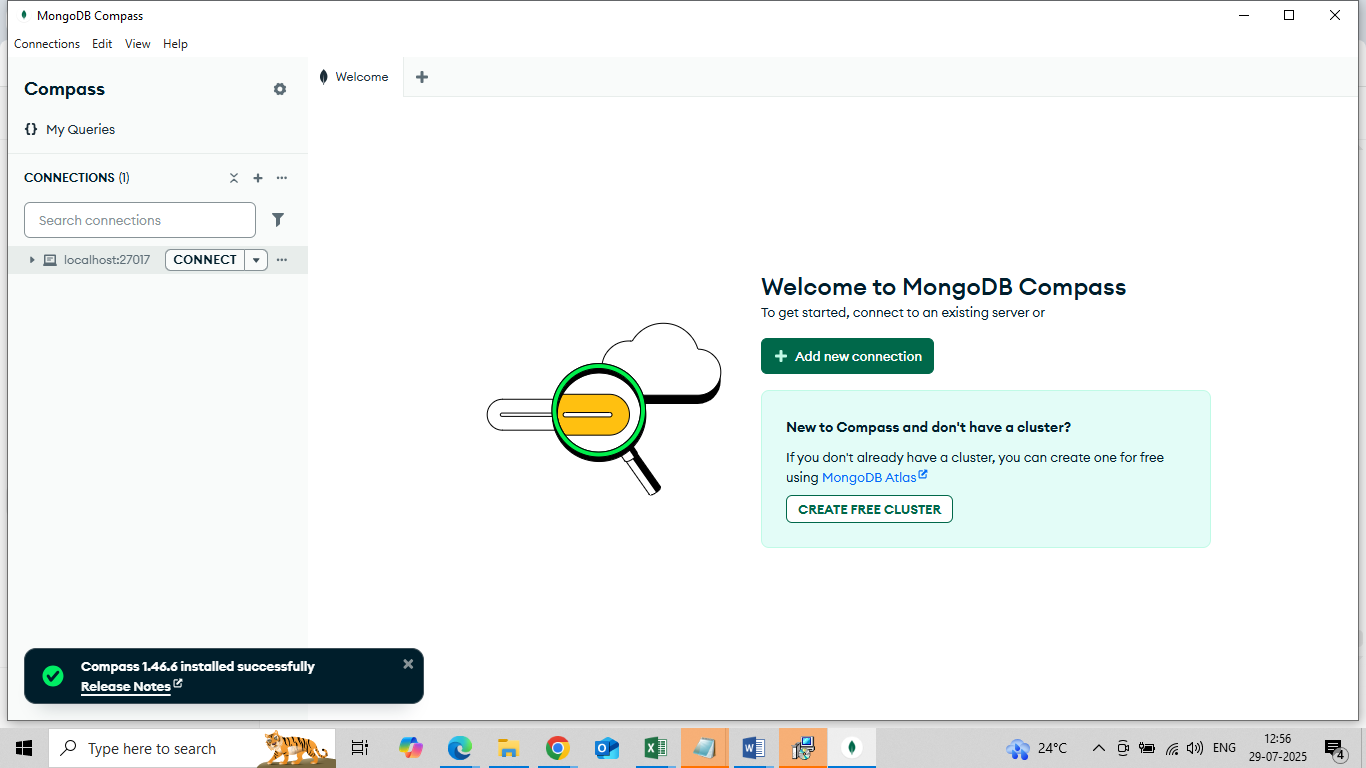
It is client like mysql command line to MongoDB server running at port 27017 (default port).

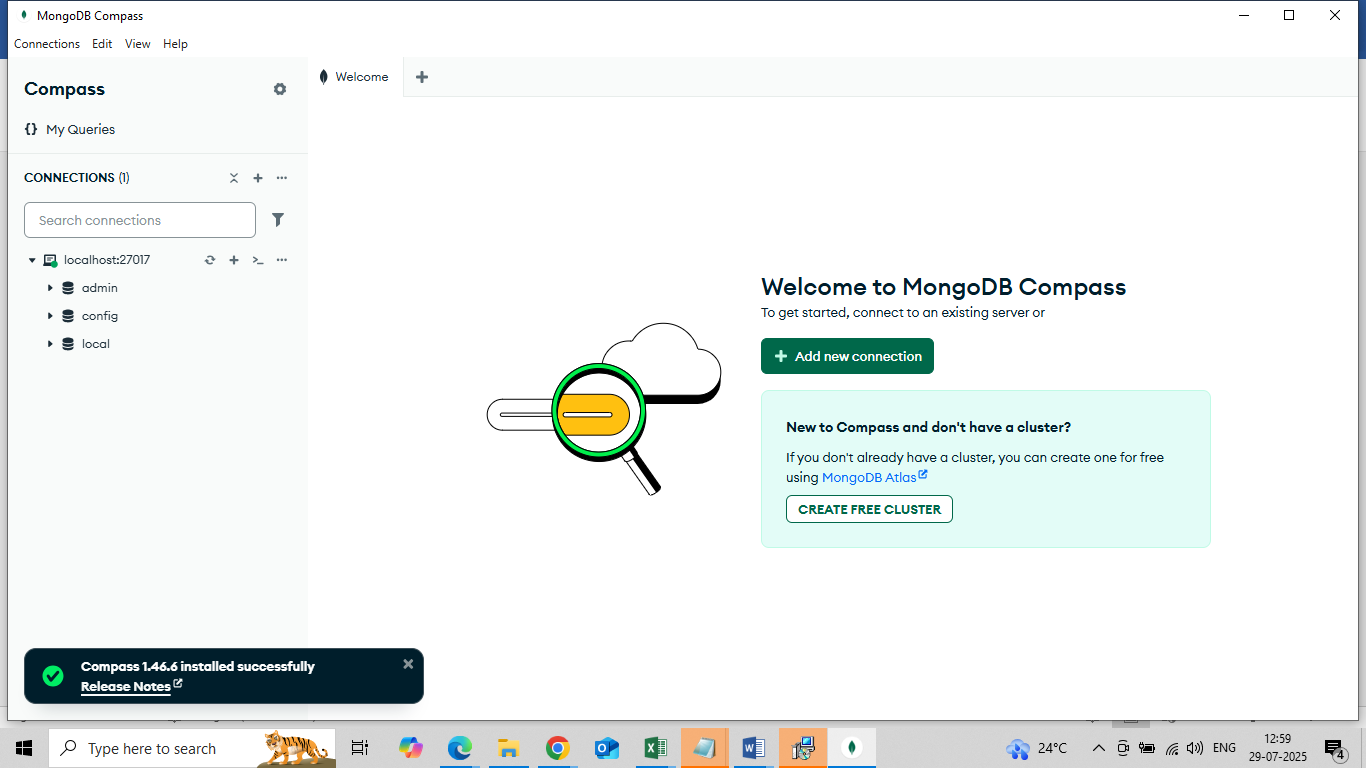
**Step 3:** Connect to MongoDB Server

MongoDB Comaps : localhost:27017 Connect button

**Click the Connect button**  
 (It should be at the bottom right of the Compass window.)

Wait a few seconds — it will connect to your local MongoDB server.





Once Connected, You will see A list of databases:

* admin
* local
* config

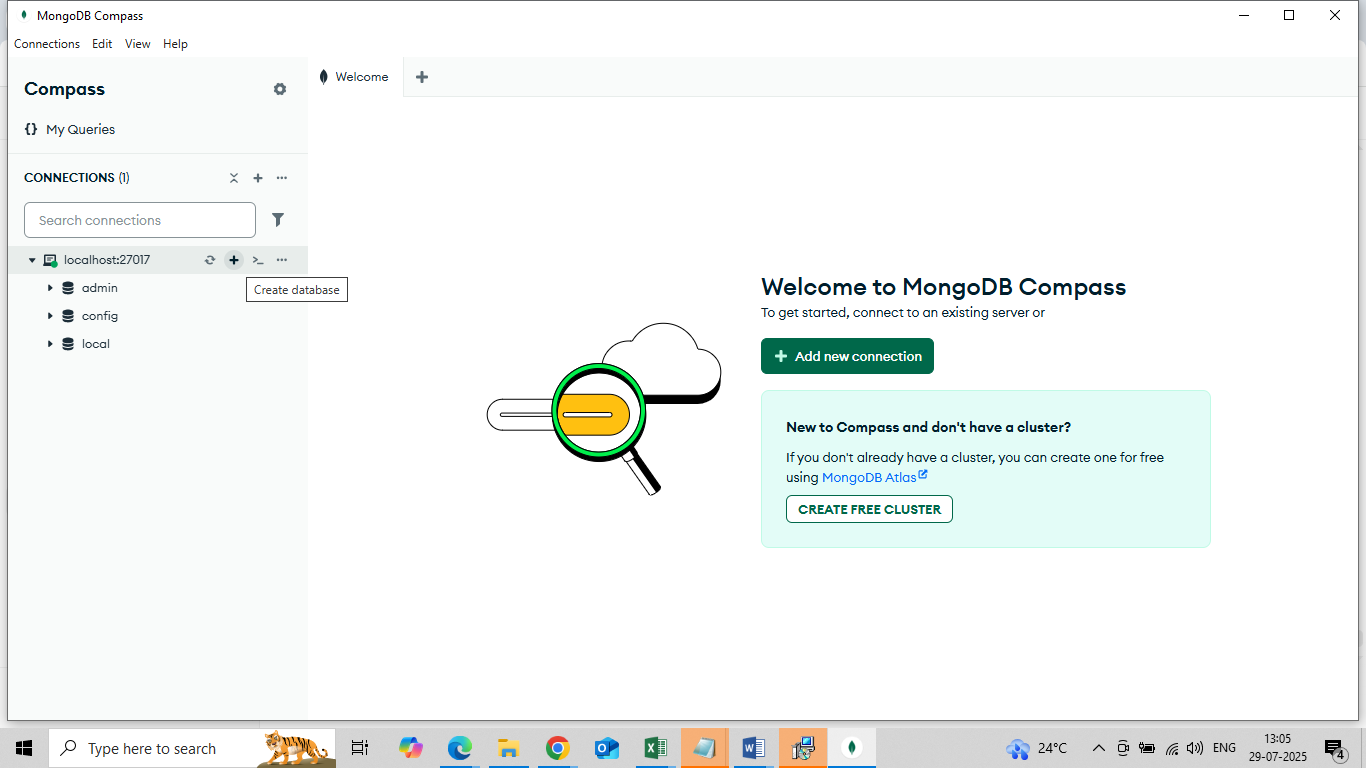
**Step 3:** Create a simple database

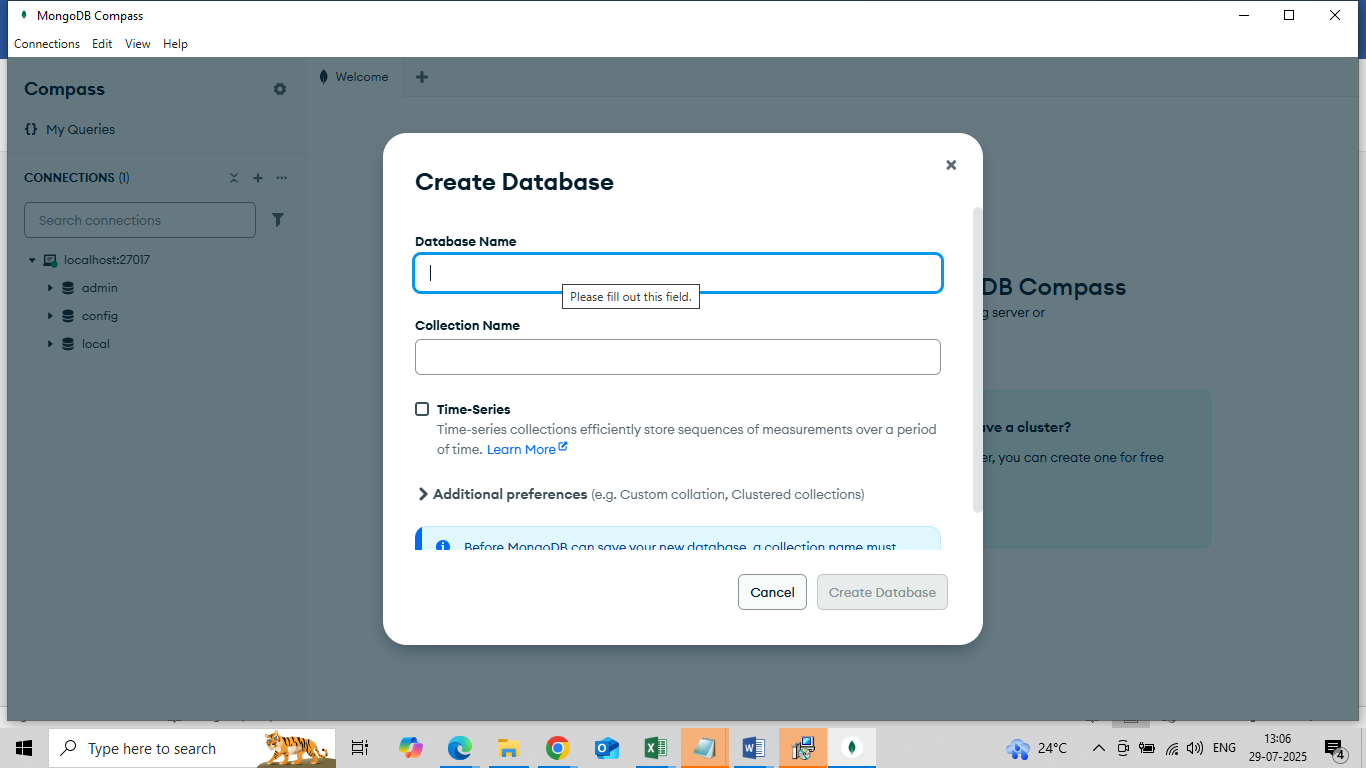
Click **“Create Database”** in the left-hand menu.

Fill In

**Database Name:** university

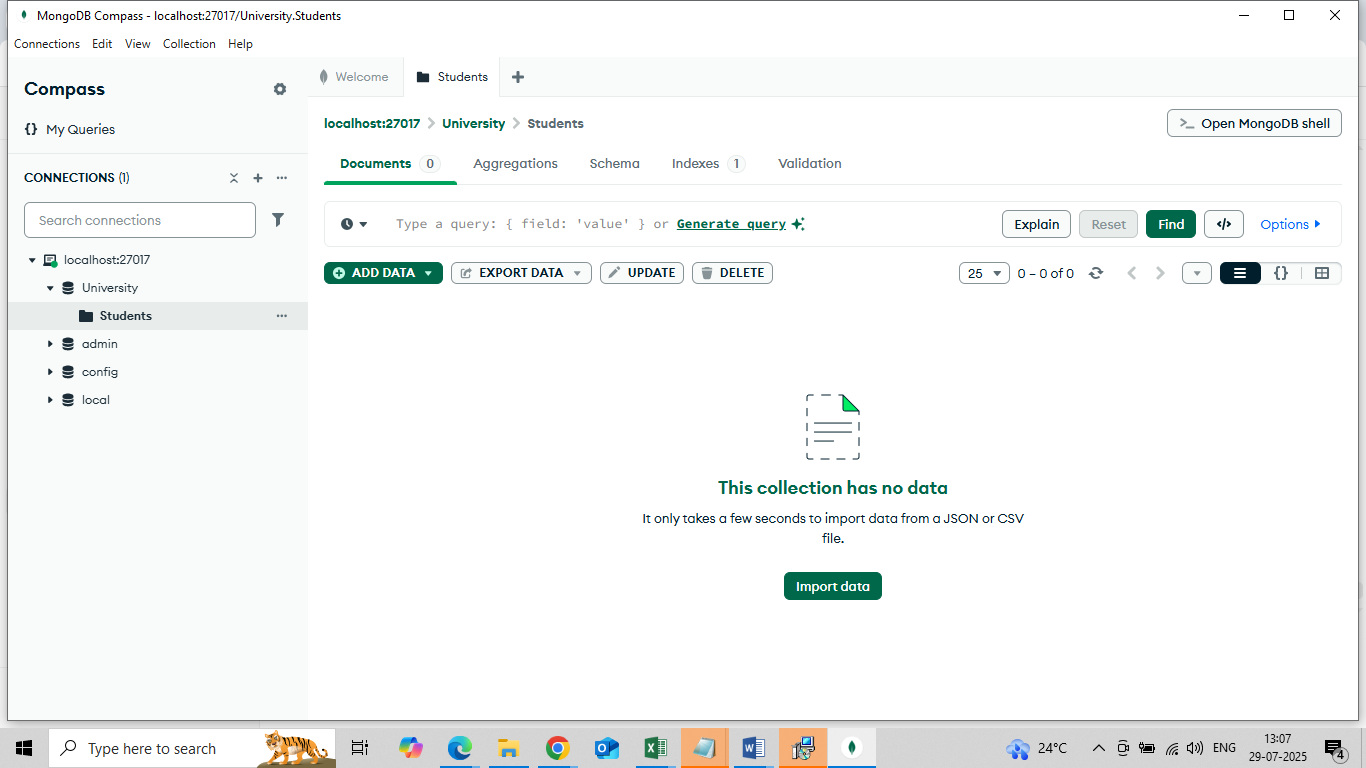
**Collection Name:** students





**Click "Time series" only if**:

* You're working with **time-stamped data**, such as:
  + Sensor data (IoT)
  + Logs with timestamps
  + Stock prices or weather data over time



**a) Insert Sample Document:**

{

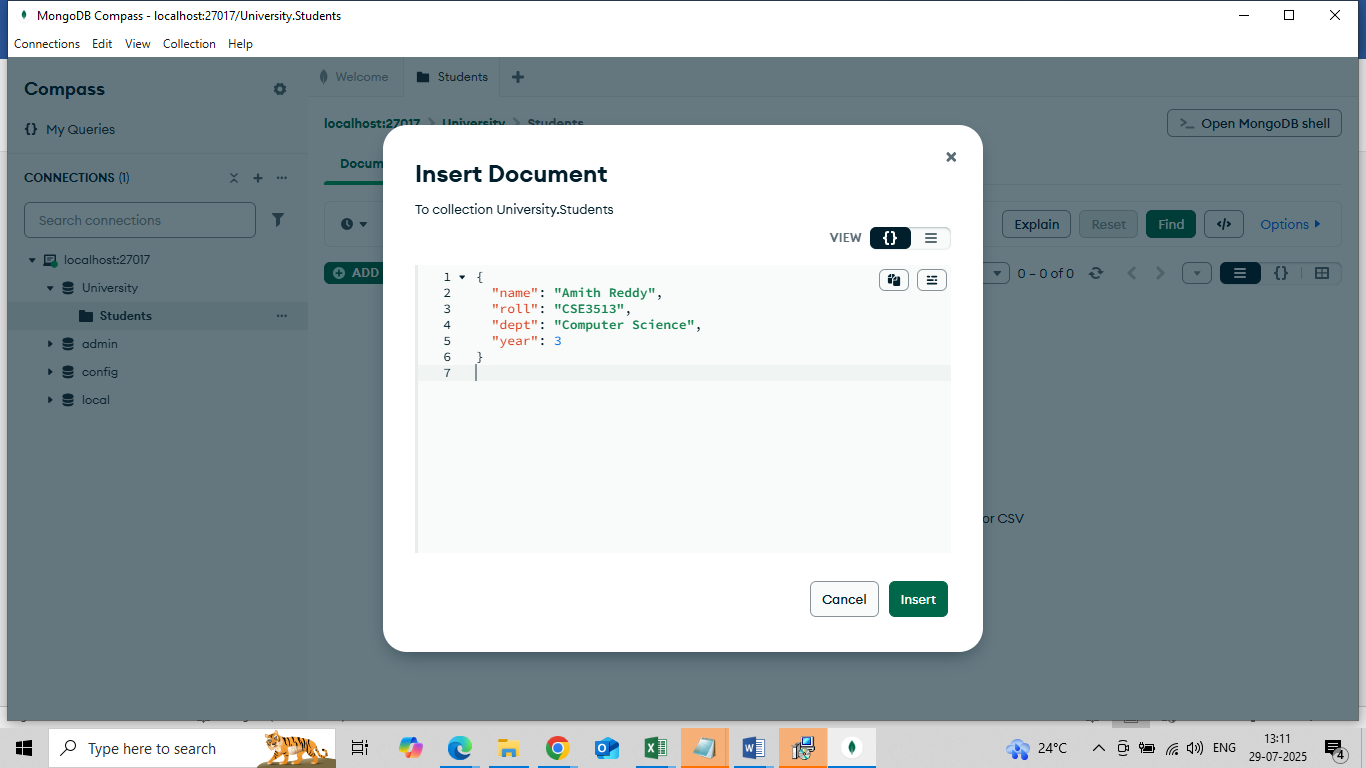
"name": "Amith Reddy",

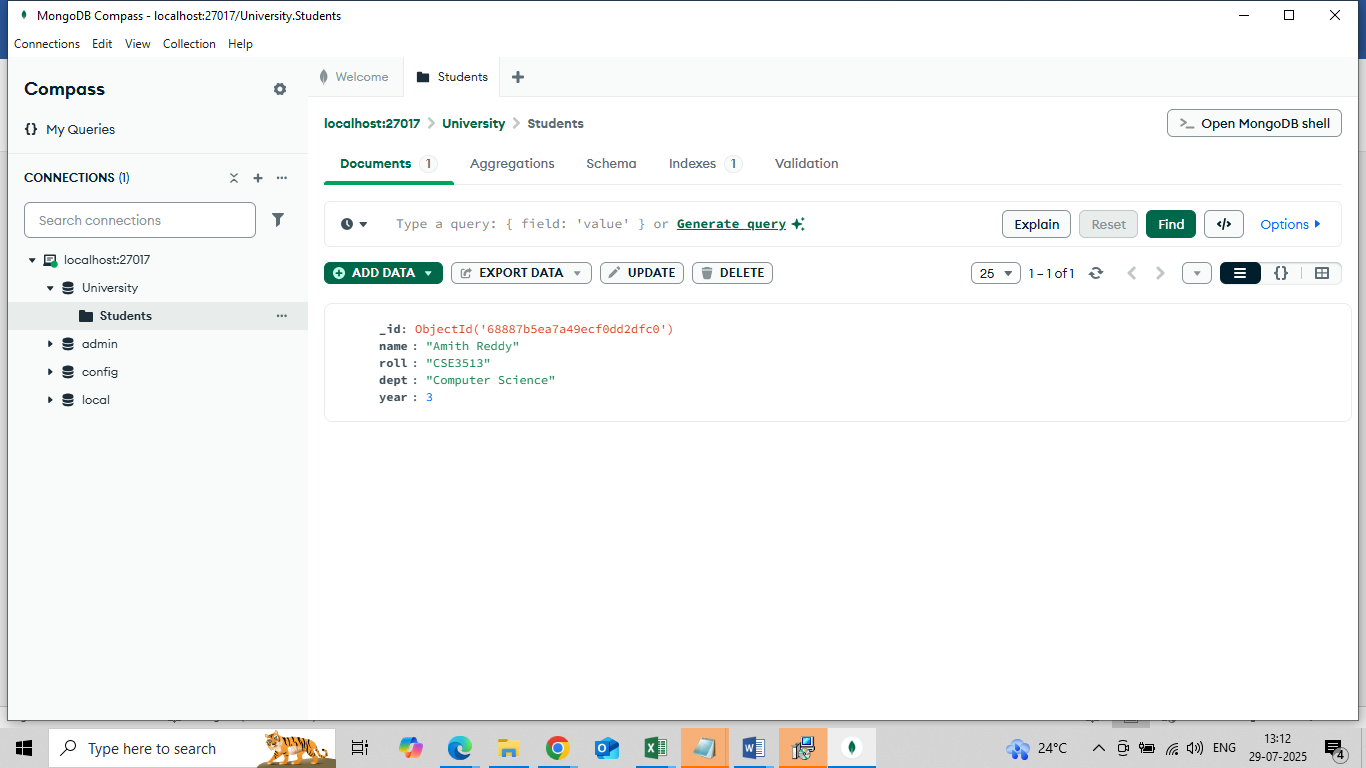
"roll": "CSE3513",

"dept": "Computer Science",

"year": 3

}

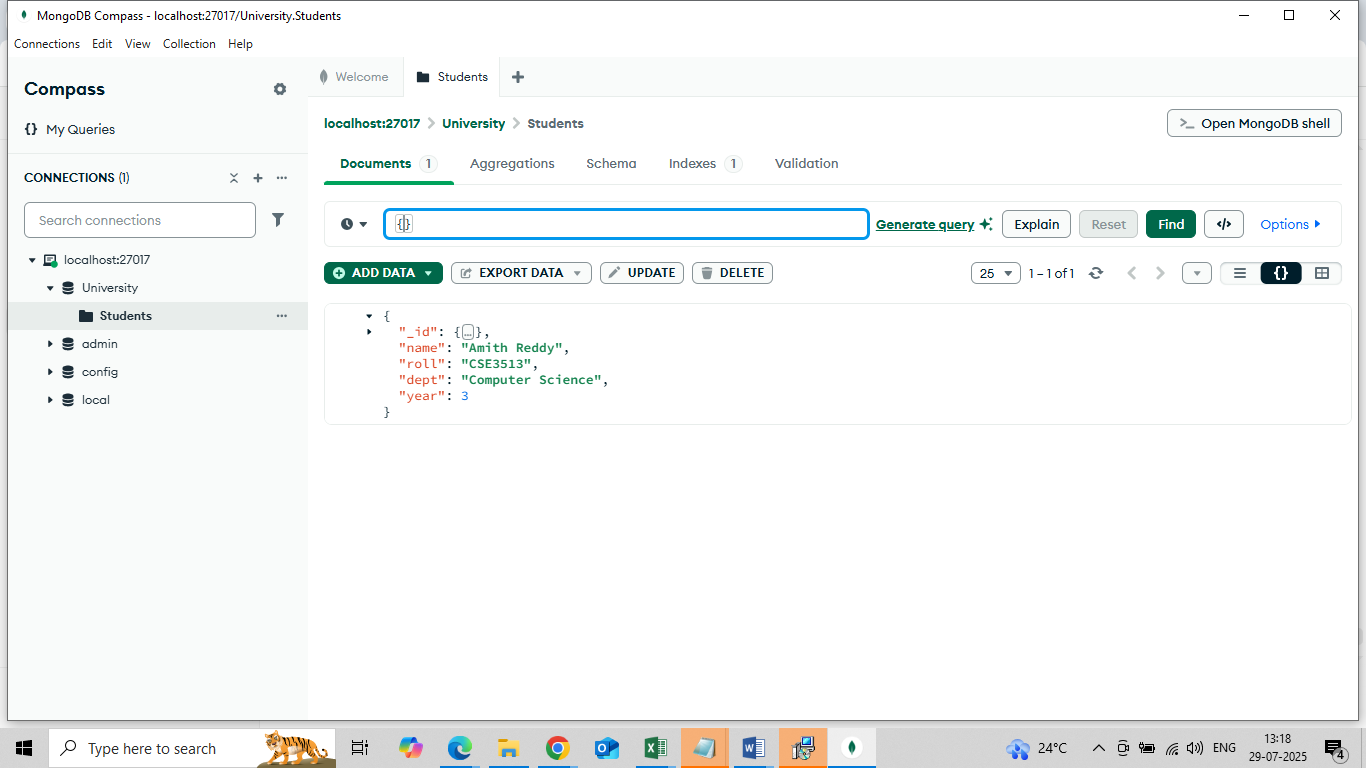




**b) View and Explore your data:**

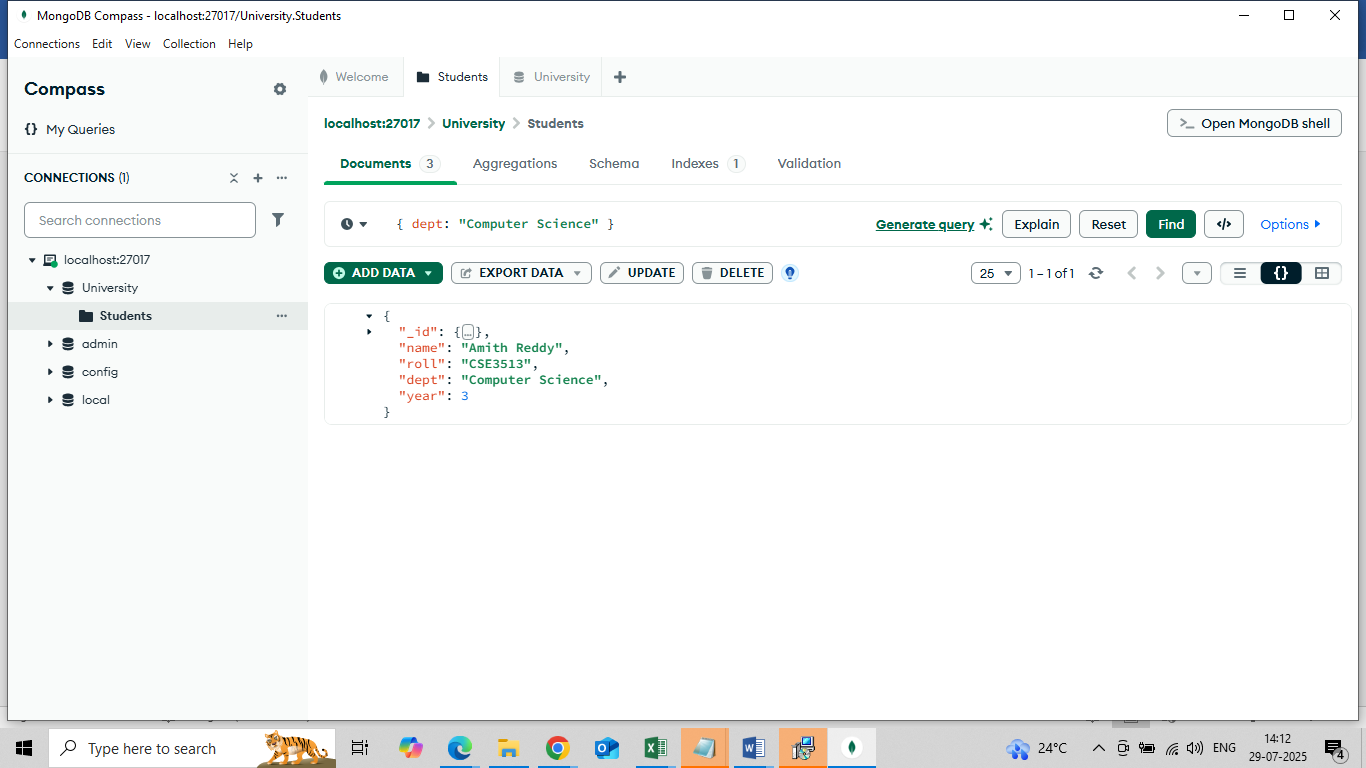
You should see your inserted student document listed under the students collection.

Use the **filter/search bar** to try simple queries, for example:

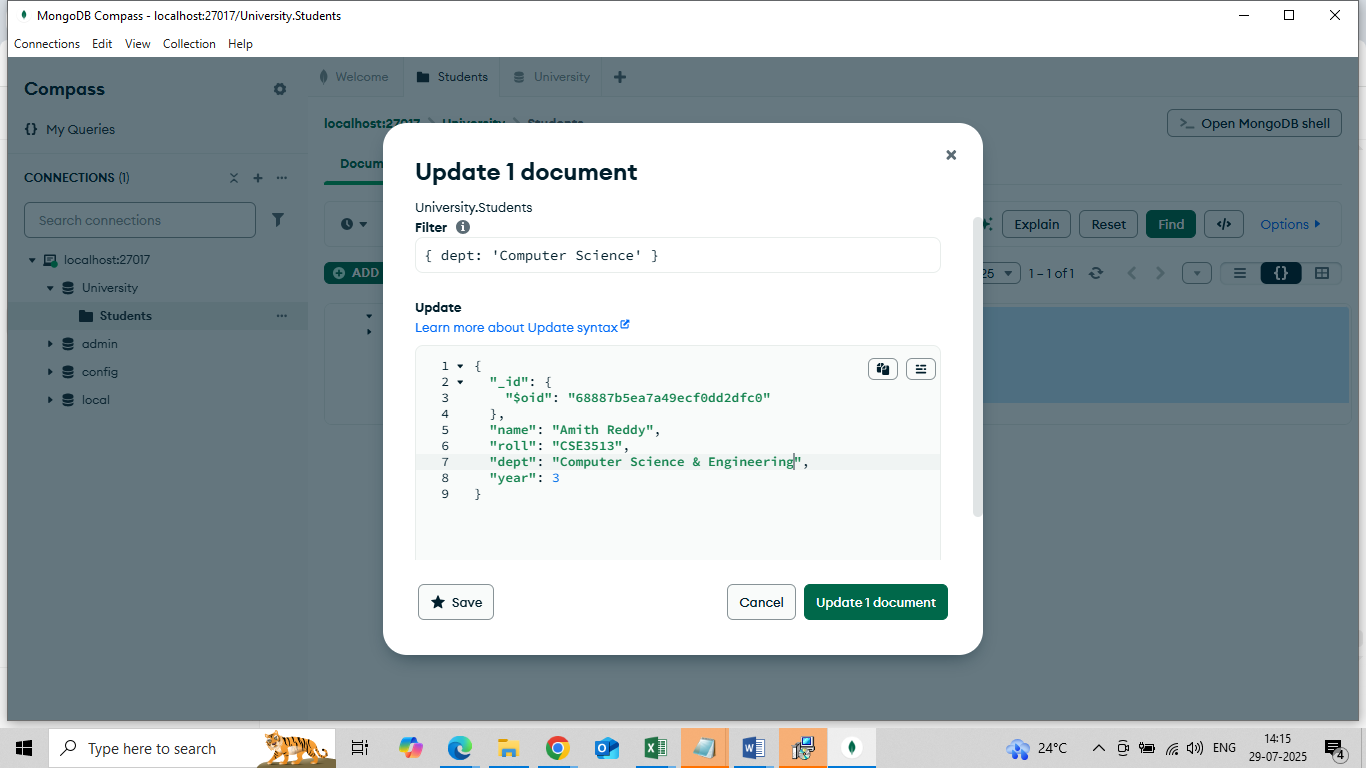


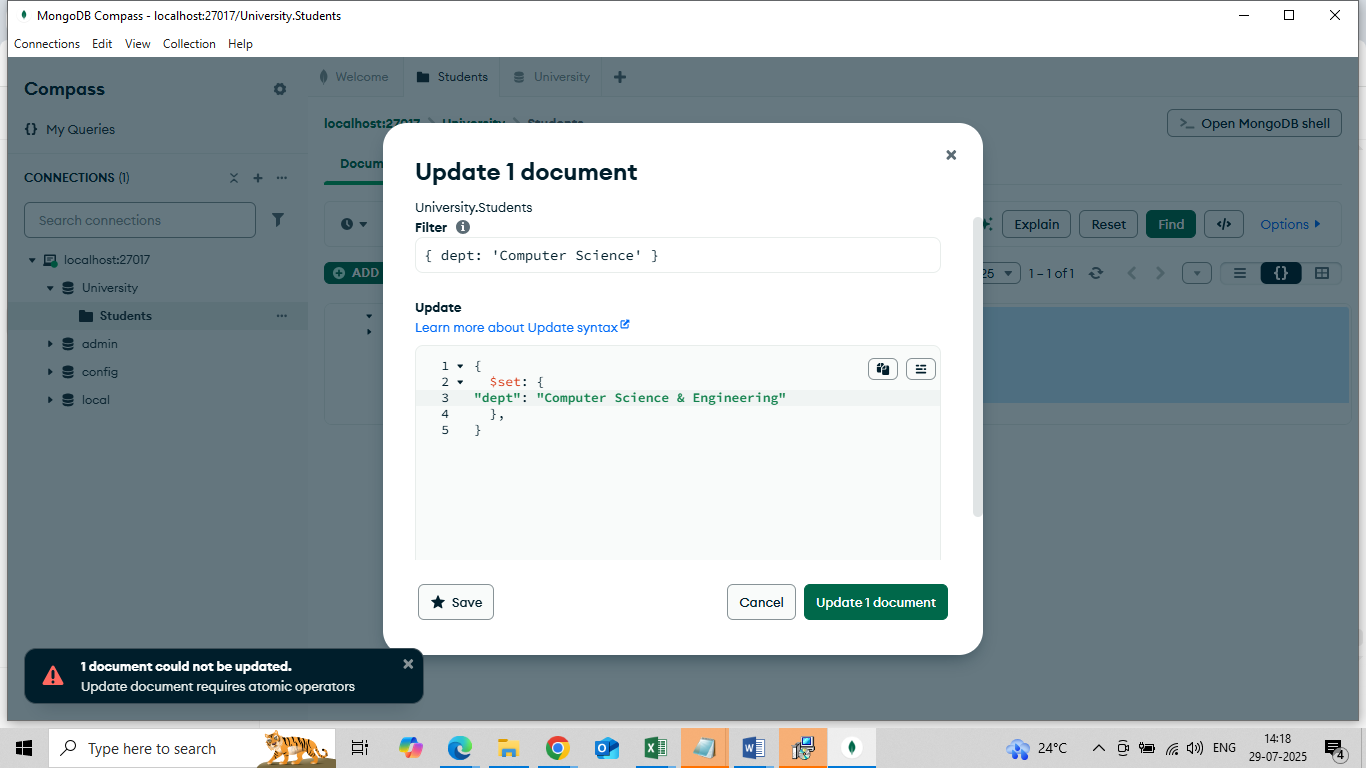
Type { "dept": "Computer Science" }

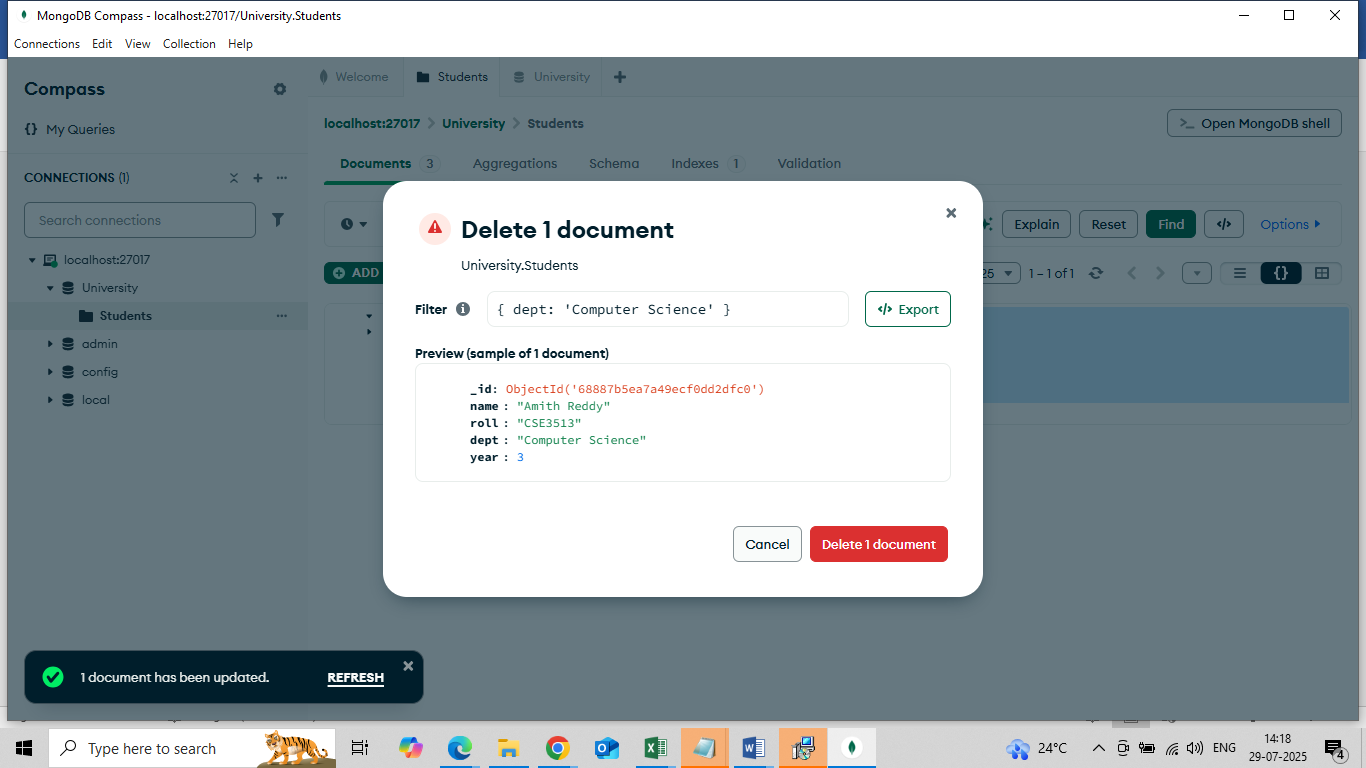
**Click the “Find” Button** (or press Enter)

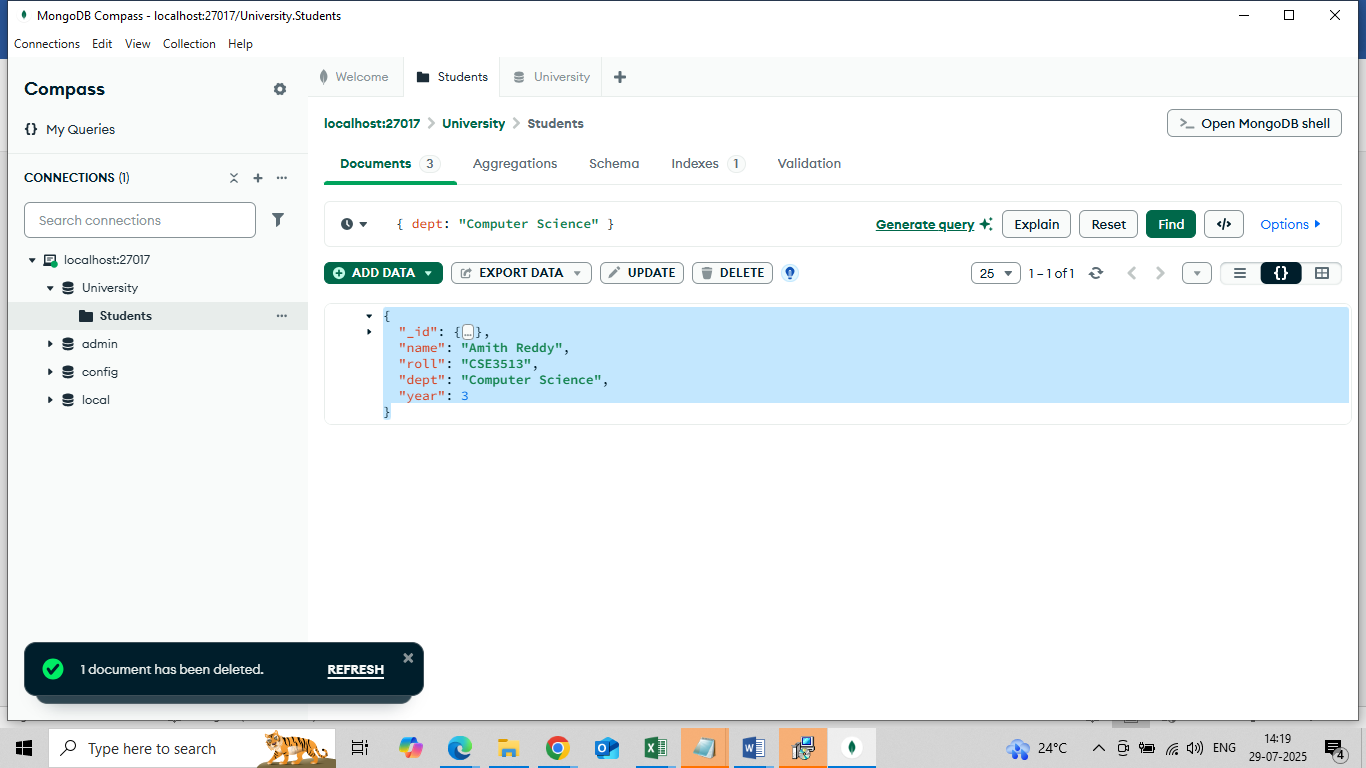


**c) Update and delete document:**







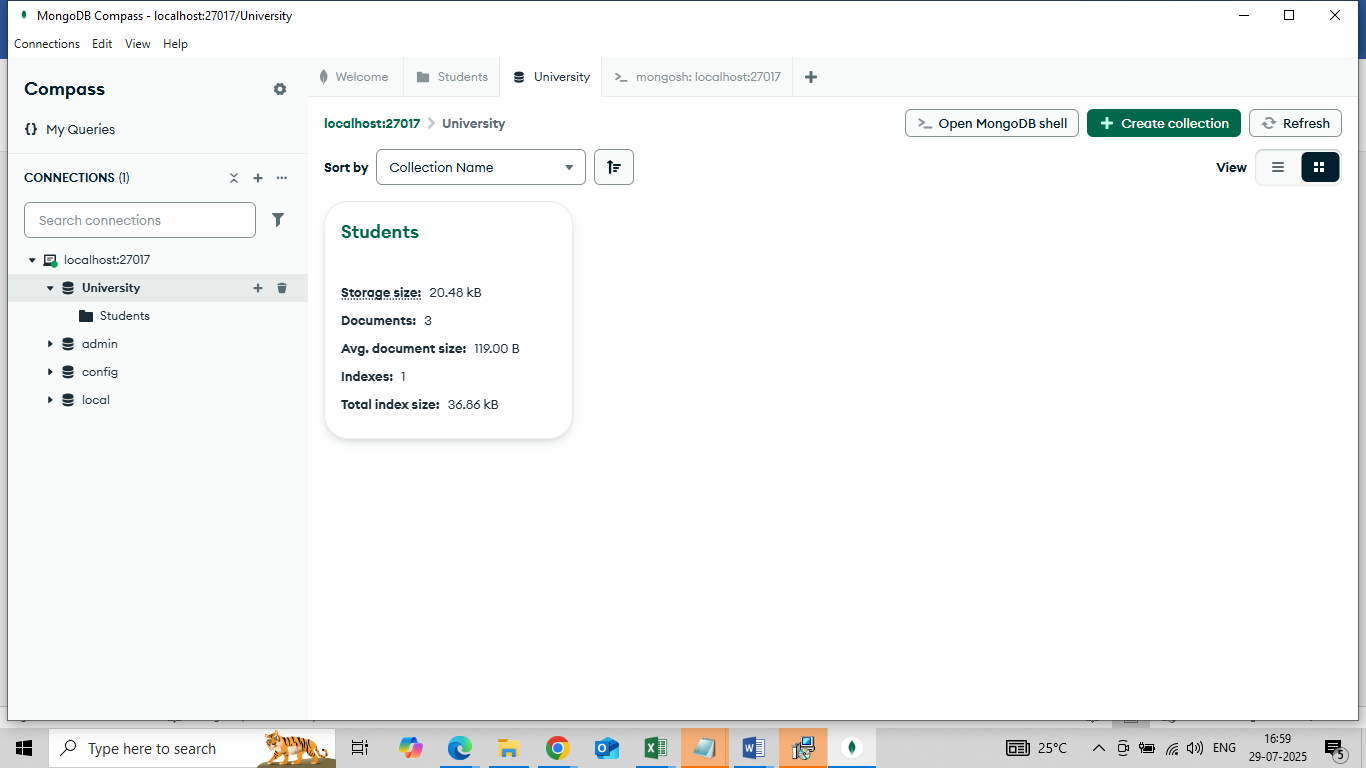


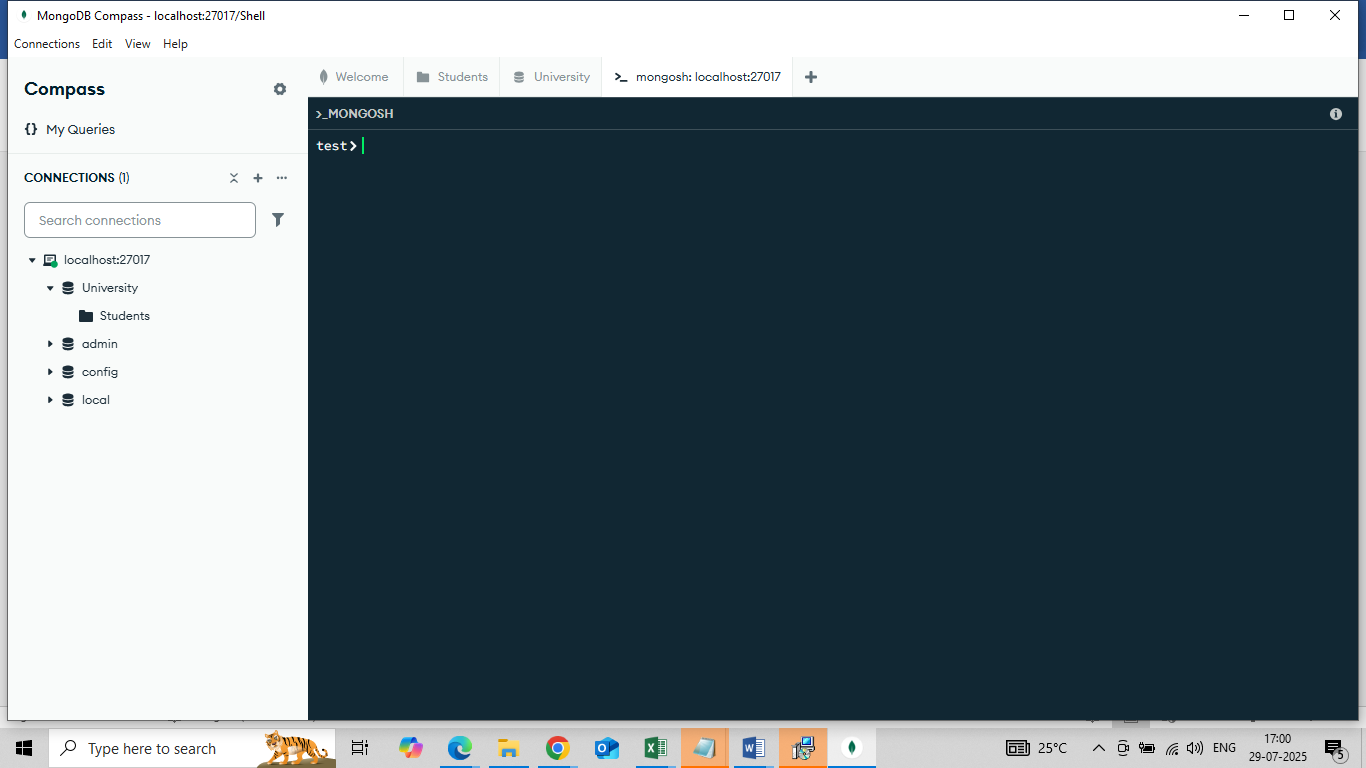
**Open and Use mongosh in Compass**

This is terminal to write nosql query for mongodb .

**Step 1: Open mongosh in Compass**

1. Open MongoDB Compass.
2. On the **left sidebar**, look for a **terminal icon** 🖥️ or a tab that says **"Command Line"** or **"Shell"**.
3. Click it — this opens the **mongosh console**. (open MongoDB Shell)





**Step 2: Basic Query Handson**

1. Use database university (just created above)

Command : use university

1. insert a student document

db.Students.insertOne({

studentID: "S101",

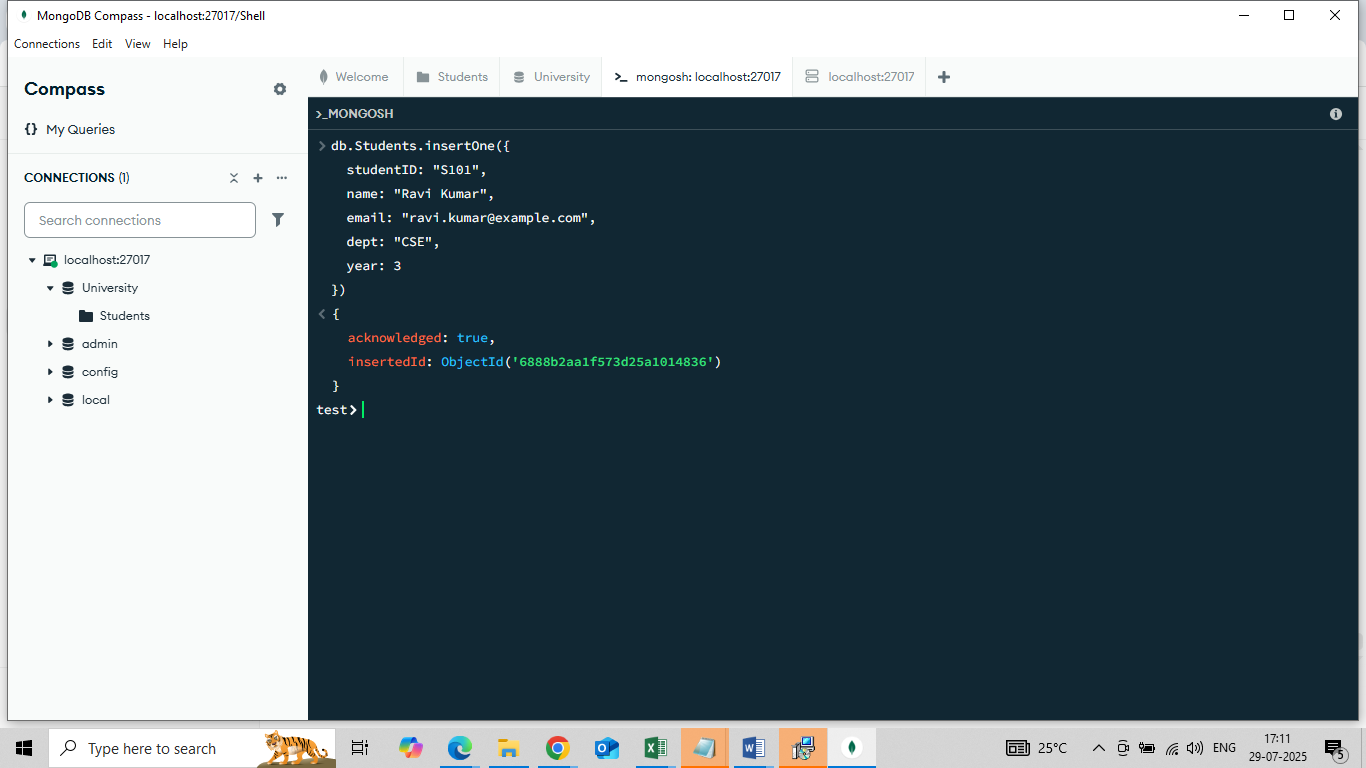
name: "Ravi Kumar",

email: "ravi.kumar@example.com",

dept: "CSE",

year: 3

})



1. Add more students
2. Check all Student

Command : db.Students.find().pretty()

1. Check a specific student

Command: db.Students().find({name:”Sunil”}).pretty()

1. Update a student

db.Students.updateOne(

{ name: "Sunil" },

{ $set: { dept: "ECE" } }

)

1. Delete a student

db.Students.deleteOne(

{ name: "Amit" }

)

# Lab challenge 1: University Management System

#### **Scenario:**

A university wants to manage its student records using MongoDB. The system should support inserting new student details, viewing them based on various filters, updating information like contact number, and deleting records of graduated students.

### 🏁 **Learning Objectives (LOs):**

* Apply CRUD operations in MongoDB Compass.
* Use filtering techniques to retrieve documents.
* Perform atomic updates using update operators.
* Use GUI-based document management via MongoDB Compass.

### 📋 **Requirements:**

#### 🧑‍🎓 Student Entity

* **Attributes**:
  + student\_id: Unique identifier
  + name: Full name
  + department: e.g., CSE, ECE, MECH
  + email: Contact email
  + phone: Contact number
  + year: Current year of study
  + cgpa: Cumulative GPA

#### ✅ Tasks:

1. **Create Operation**
   * Use MongoDB Compass to:
     + Create a new **database** called university
     + Create a **collection** named students
     + Insert at least **5 student documents** with the given fields.
2. **Read Operation**
   * Retrieve:
     + A student by student\_id
     + All students in a specific department (e.g., ECE)
     + Students with cgpa greater than 8.0
3. **Update Operation**
   * Update the phone number of a student using student\_id
   * Increase the cgpa of a student by 0.1 using the $inc operator
4. **Delete Operation**
   * Delete a student who has graduated (e.g., year: 4)
   * Delete a student using student\_id

### 🧠 **Real-Time Scenario**

You are the administrator in charge of academic records. As new students join, you add them to the system. If a student updates their phone number or improves their CGPA, their record is updated accordingly. When students graduate, their records are removed from the database.

Happy Learning