

# Workforce Management System

## Milestone: Implementation in NoSQL

Group 20

Student1 Ankita Shukla

Student 2 Soham Palnitkar

857-498-9754 (Tel of Student 1)

857-693-8702 (Tel of Student 2)

shukla.ank@northeastern.edu

palnitkar.s@northeastern.edu

Percentage of Effort Contributed by Student1: \_\_\_50\_\_\_

Percentage of Effort Contributed by Student2: \_\_\_50\_\_\_

Signature of Student 1: Ankita Shukla\_\_\_\_\_

Signature of Student 2: Soham Palnitkar\_\_\_\_\_

Submission Date: \_\_\_\_\_12/03/2022\_\_\_\_\_

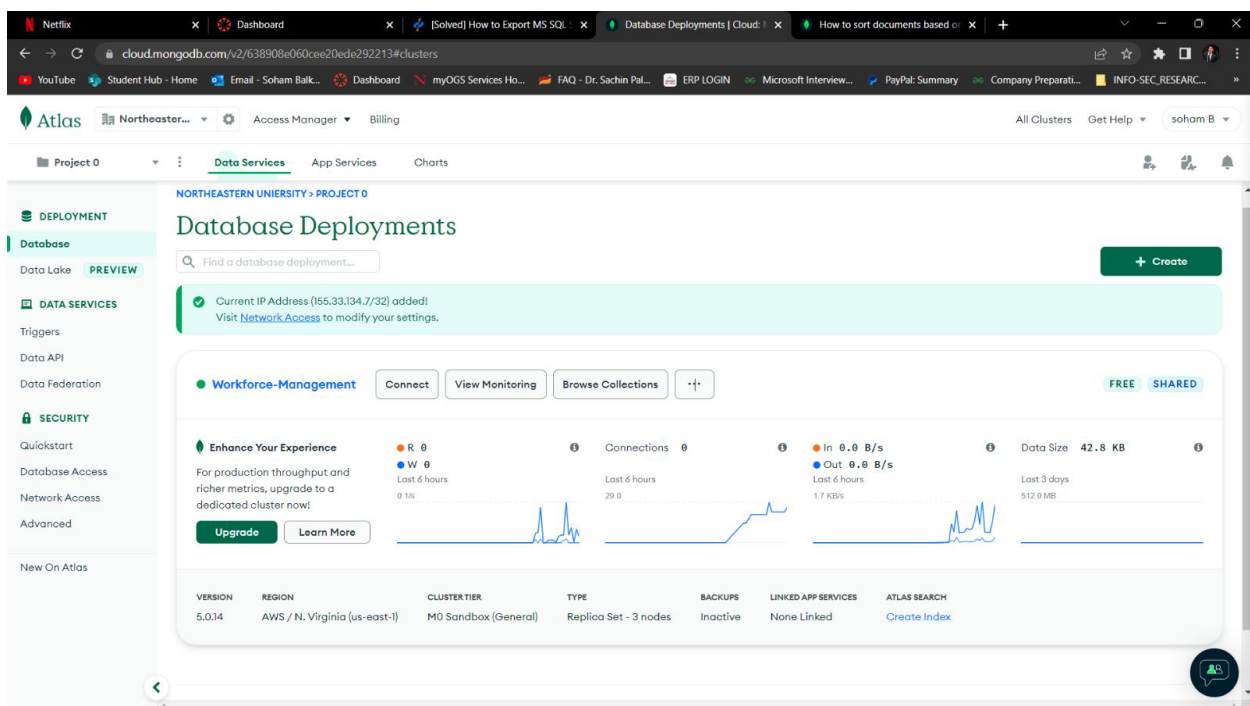
# Implementation in NoSQL

**NoSQL database** technology stores information in JSON documents instead of columns and rows used by relational databases. To be clear, NoSQL stands for “not only SQL” rather than “no SQL” at all.

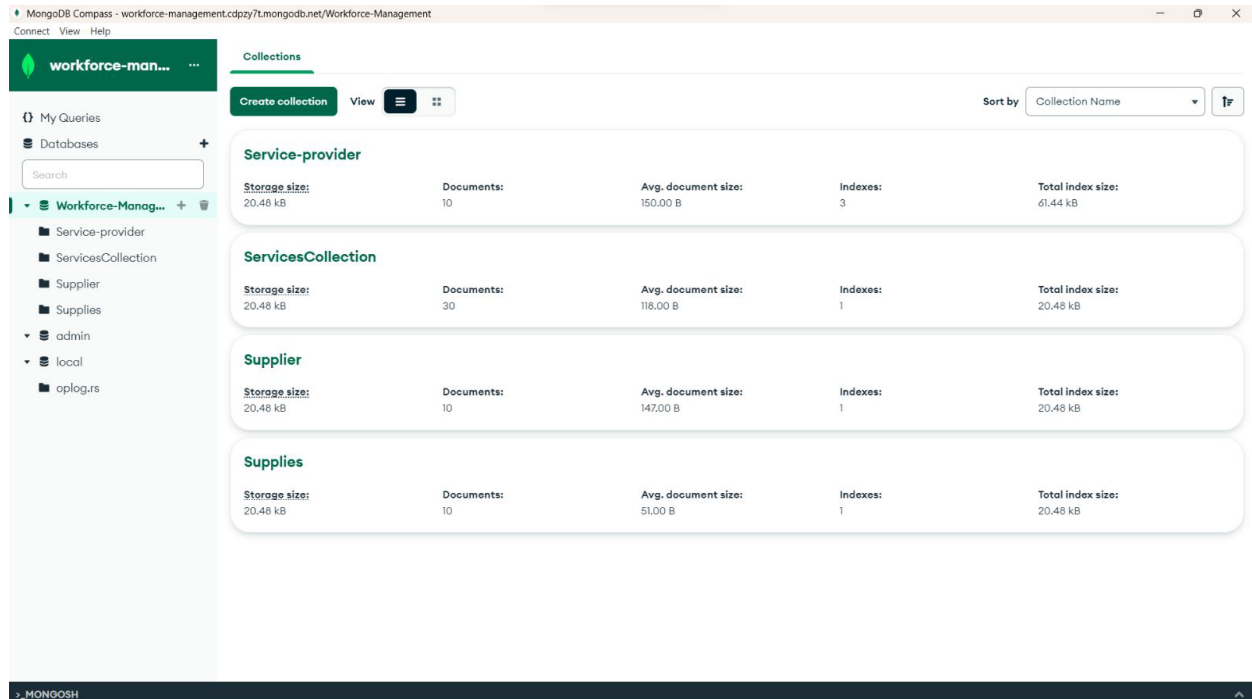
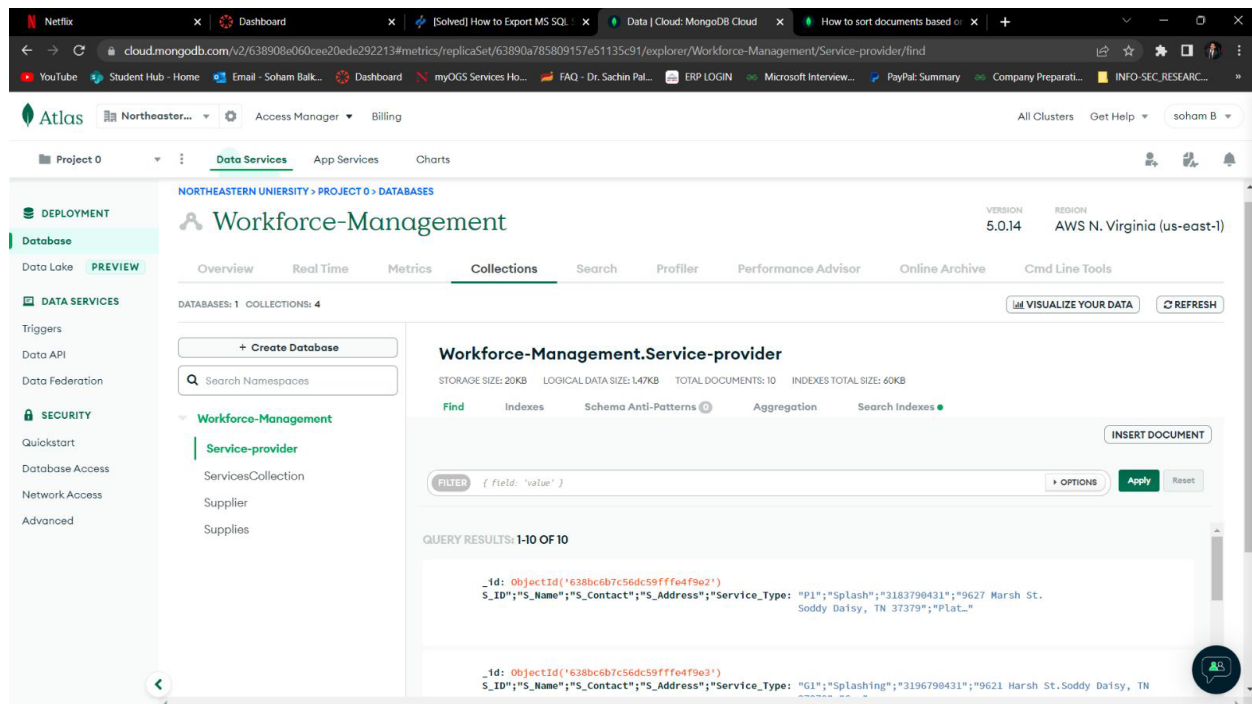
Combines the flexibility of JSON with the power of SQL for the best of both worlds. Consequently, NoSQL databases are built to be flexible, scalable, and capable of rapidly responding to the data management demands of modern businesses.

**MongoDB** is one of the foremost open-source NoSQL systems. It is a document-oriented database that uses dynamic schemas to store JSON-like documents. This database solution features a flexible data model, enabling users to store unstructured data. Users can also access full indexing support and replication through intuitive API.

The following screenshot represents the deployment for Workforce Management System in MongoDB.



The following screenshots represent the collections created under the Workforce Management System in MongoDB.



Below are the examples of query implementation on Workforce Management System in MongoDB.

1. The following screenshot represents the output for selecting supplier details for all those suppliers who supply 'standard' paint type for Workforce Management System in MongoDB.

The screenshot shows the MongoDB Compass interface. The database is 'workforce-man...' and the collection is 'Workforce-Mana...'. The query filter is set to `{Paint_Type: "Standard"}`. The results show four documents, each representing a supplier with details like `_id`, `Sup_ID`, `Paint_Type`, `Sup_Contact`, and `Sup_Address`.

Supplier ID	Paint Type	Supplier Contact	Supplier Address
638bd26bc56dc59fffe4fa2f	Standard	6072201152	9418 North Indian Summer Court, Evansville, IN 47711
638bd26bc56dc59fffe4fa30	Standard	8557989504	75 Thompson Dr., Absecon, NJ 08205
638bd26bc56dc59fffe4fa31	Standard	2062360859	75 North Kent Drive, North Canton, OH 44720
638bd26bc56dc59fffe4fa33	Standard	7484534730	506 Tailwater St., Morton Grove, IL 60053

2. The following screenshot represents the output for selecting supplier details for all those suppliers who supply 'premium' paint type for Workforce Management System in MongoDB and sort the result in descending order of supplier Id.

The screenshot shows the MongoDB Compass interface. The database is 'workforce-man...' and the collection is 'Workforce-Mana...'. The query filter is set to `{Paint_Type: "Premium"}`. The results are sorted by `Sup_ID` in descending order. The results show three documents, each representing a supplier with details like `_id`, `Sup_ID`, `Paint_Type`, `Sup_Contact`, and `Sup_Address`.

Supplier ID	Paint Type	Supplier Contact	Supplier Address
638bd26bc56dc59fffe4fa34	Premium	2245392775	580 South Fairfield Ave., Scotch Plains, NJ 07076
638bd26bc56dc59fffe4fa2d	Premium	3276222045	688 W. St. Louis St., Glastonbury, CT 06033
638bd26bc56dc59fffe4fa2e	Premium	6102170630	9260 SW. Jefferson St., Hoboken, NJ 07030

3. The following screenshot represents the output for implementing aggregations and create a pipeline for Workforce Management System in MongoDB, displaying Service Provider Id, Service Provider name for service providers who's service type is gold.

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'workforce-man...' with collections 'Suppliers', 'admin', 'local', and 'oplog.rs'. The main panel shows the 'Workforce-Management.ServicesCollection' with 30 documents and 1 index. An aggregation pipeline is defined with a single stage: '\$match' with the criteria '{Service\_Type: "Gold"}'. The results show four documents:

_id	S_ID	S_Name	Service_Type
ObjectId('638bd258c56dc59ffe4fa23')	G1	Splashing	Gold
ObjectId('638bd258c56dc59ffe4fa24')	G2	g1m	Gold
ObjectId('638bd258c56dc59ffe4fa2a')	G4	Hallo	Gold
ObjectId('638bd258c56dc59ffe4fa2b')	G5	Hike	Gold