

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ЗВІТ З ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«СУЧАСНІ ПАРАДИГМИ ЗБЕРЕЖЕННЯ ДАНИХ»

Виконав: студент групи КН-22-1(а)

Панцюк К.В.

Перевірив: Андрєєв П.І.

КРЕМЕНЧУК 2025

Лабораторна робота № 1

Тема. Основи роботи з MongoDB

Мета роботи: Ознайомитися з основами роботи із документно-орієнтованою базою даних MongoDB. Навчитися створювати бази даних і колекції, додавати документи, за допомогою базових команд MongoDB.

Хід виконання роботи

Пункт 1

1. Встановити MongoDB Community Edition (локально) та MongoDB Shell. Потрібно перейти на офіційний сайт MongoDB, обрати свою операційну систему (Windows, macOS або Linux) та завантажити інсталятор.

Після встановлення:

- Перевірити, чи MongoDB працює: `mongod --version`
- У новій вкладці терміналу відкрити Mongo Shell: `mongosh`

Виконання пункту 1

```
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\panda>mongod --version
db version v8.0.13
Build Info: {
  "version": "8.0.13",
  "gitVersion": "8dc5cd2a30c4524132e2d44bb314544dc477e611",
  "modules": [],
  "allocator": "tcmalloc-gperf",
  "environment": {
    "distmod": "windows"
  }
}

C:\Users\panda>mongosh
Current Mongosh Log ID: 68b9767b7b5b5db777735188
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.7
Using MongoDB: 8.0.13
Using Mongosh: 2.5.7

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-09-04T13:10:45.011+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> |
```

Рисунок 1.1–Результат виконання пункту 1

У ході виконання цього пункту мною було встановлено MongoDB Community Edition та MongoDB Shell після чого було виконано 2-і команди `mongod --version`, яка демонструє саму версію встановленого серверу та виконано команду `mongosh`, яка демонструє чи успішно підключено до серверу, саму версію сервера та версію shell'а.

Пункт 2

2. Створити базу students з колекцією users

У mongosh потрібно ввести згідно варіанту завдання декілька документів,

наприклад:

use students

```
db.users.insertMany([
  {
    name: "Іван Іванов",
    age: 20,
    group: "CS-101",
    grades: [90, 85, 78]
  },
  {
    name: "Марія Петренко",
    age: 19,
    group: "CS-101",
    grades: [88, 92, 81]
  },
  {
    name: "Олег Сидоренко",
    age: 21,
    group: "CS-102",
    grades: [70, 75, 80]
  }
])
```

Виконання пункту 2

```
test> use students
switched to db students
students> show collections
users
students> db.users.find().pretty()
[
  {
    _id: ObjectId('68b979aa8da13c726e735189'),
    name: 'Іван Іванов',
    age: 20,
    group: 'CS-101',
    grades: [ 90, 85, 78 ]
  },
  {
    _id: ObjectId('68b979aa8da13c726e73518a'),
    name: 'Марія Петренко',
    age: 19,
    group: 'CS-101',
    grades: [ 88, 92, 81 ]
  },
  {
    _id: ObjectId('68b979aa8da13c726e73518b'),
    name: 'Олег Сидоренко',
    age: 21,
    group: 'CS-102',
    grades: [ 70, 75, 80 ]
  }
]
```

Рисунок 1.2—Результат створення через командний рядок

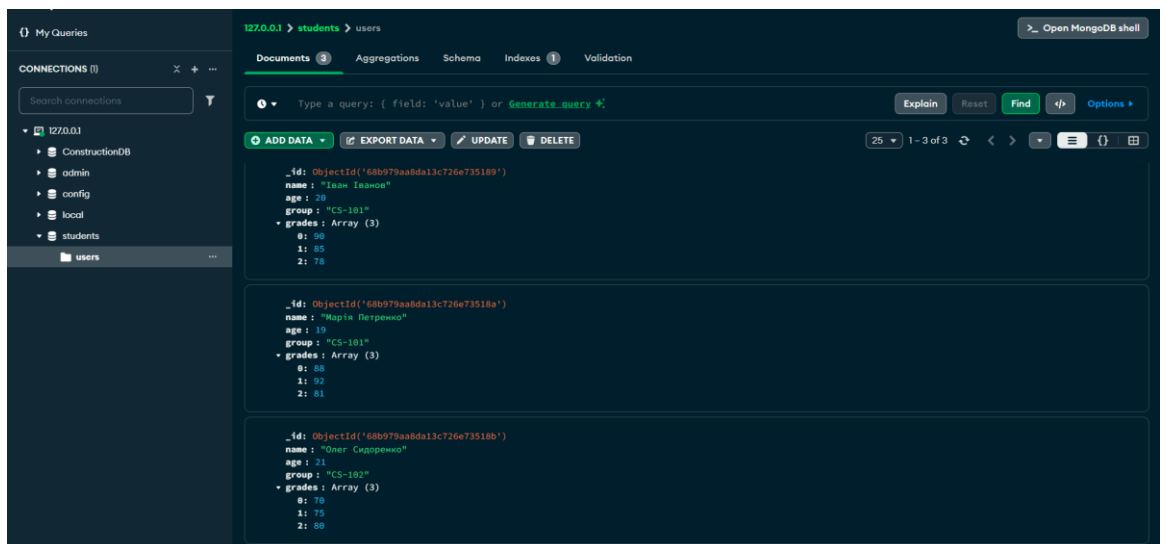


Рисунок 1.3—Відображення даних у MongoDB Compass

При виконанні цього пункту мною було виконано створення :

–База даних: students;

–Колекція: users;

–Документи: 3 окремі документи у форматі BSON.

Пункт 3

3. Створити адміністратора та користувача з частковим доступом.

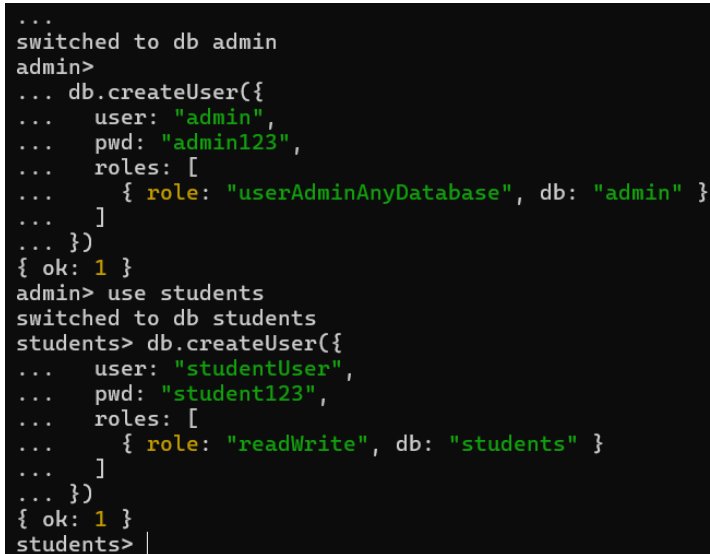
Запустити mongosh та згідно варіанту завдання створити адміністратора, наприклад:

```
mongosh
use admin
db.createUser({
  user: "admin",
  pwd: "admin123",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
})
```

Створення обмеженого користувача для бази students, наприклад:

```
use students
db.createUser({
  user: "studentUser",
  pwd: "student123",
  roles: [ { role: "readWrite", db: "students" } ]
})
```

Виконання пункту 3



```
...
switched to db admin
admin>
... db.createUser({
...   user: "admin",
...   pwd: "admin123",
...   roles: [
...     { role: "userAdminAnyDatabase", db: "admin" }
...   ]
... })
{ ok: 1 }
admin> use students
switched to db students
students> db.createUser({
...   user: "studentUser",
...   pwd: "student123",
...   roles: [
...     { role: "readWrite", db: "students" }
...   ]
... })
{ ok: 1 }
students> |
```

Рисунок 1.4–Результат виконання пункту 3

Я створив адміністратора в базі admin з роллю userAdminAnyDatabase — це дає йому повні права на керування користувачами в будь-якій базі.

Після чого створив користувача studentUser в базі students з роллю readWrite — він має доступ лише до читання та запису в межах цієї бази.

Обидві команди завершилися успішно ({ ok: 1 }), що означає, що MongoDB прийняла їх без помилок.

Пункт 4

4. Увімкнути автентифікацію.

1. Вийти з mongosh.

2. Відредагувати конфігураційний файл mongod.conf (зазвичай у /etc/mongod.conf або C:\Program Files\MongoDB\Server\X.X\bin\mongod.cfg).

3. Додати або змінити:

security:

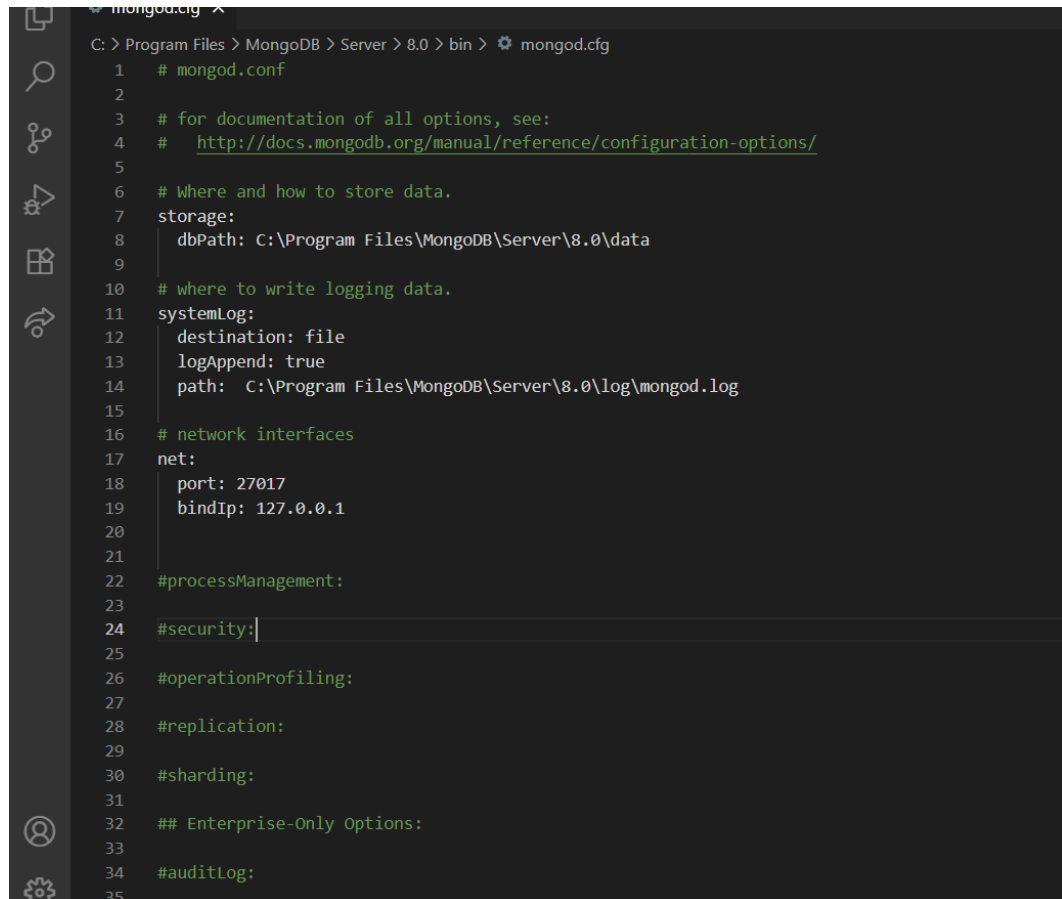
authorization: enabled

3. Перезапустити MongoDB:

sudo systemctl restart mongod # Linux

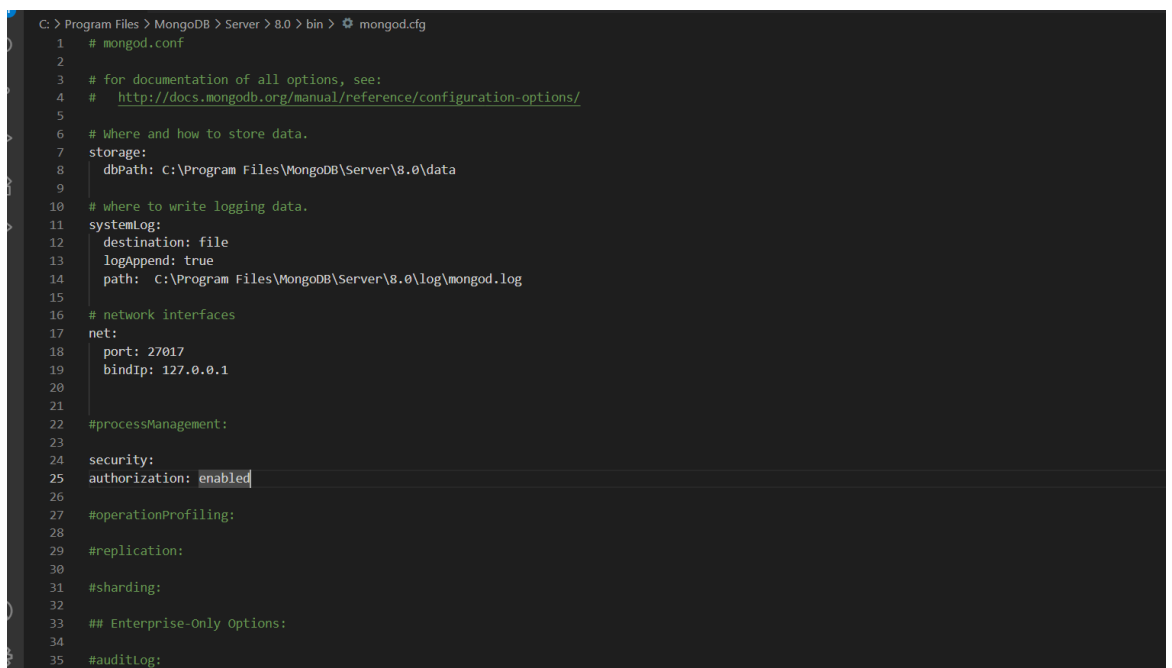
або вручну перезапустити mongod.exe на Window

Виконання пункту 4



```
C: > Program Files > MongoDB > Server > 8.0 > bin > mongod.cfg
1 # mongod.conf
2
3 # for documentation of all options, see:
4 # http://docs.mongodb.org/manual/reference/configuration-options/
5
6 # Where and how to store data.
7 storage:
8   dbPath: C:\Program Files\MongoDB\Server\8.0\data
9
10 # where to write logging data.
11 syslog:
12   destination: file
13   logAppend: true
14   path: C:\Program Files\MongoDB\Server\8.0\log\mongod.log
15
16 # network interfaces
17 net:
18   port: 27017
19   bindIp: 127.0.0.1
20
21
22 #processManagement:
23
24 #security:
25
26 #operationProfiling:
27
28 #replication:
29
30 #sharding:
31
32 ## Enterprise-Only Options:
33
34 #auditLog:
35
```

Рисунок 1.5– Початковий вигляд файлу mongod.cfg



```
C: > Program Files > MongoDB > Server > 8.0 > bin > mongod.cfg
1 # mongod.conf
2
3 # for documentation of all options, see:
4 # http://docs.mongodb.org/manual/reference/configuration-options/
5
6 # Where and how to store data.
7 storage:
8   dbPath: C:\Program Files\MongoDB\Server\8.0\data
9
10 # where to write logging data.
11 syslog:
12   destination: file
13   logAppend: true
14   path: C:\Program Files\MongoDB\Server\8.0\log\mongod.log
15
16 # network interfaces
17 net:
18   port: 27017
19   bindIp: 127.0.0.1
20
21
22 #processManagement:
23
24 security:
25   authorization: enabled
26
27 #operationProfiling:
28
29 #replication:
30
31 #sharding:
32
33 ## Enterprise-Only Options:
34
35 #auditLog:
36
```

Рисунок 1.6–Внесені зміни у файл

При виконанні таких маніпуляцій MongoDB буде вимагати автентифікацію при кожному підключенні, обмежуючи доступ лише для зареєстрованих користувачів.

Пункт 5

5. Перевірити доступ користувача.

Потрібно вийти та підключитися, наприклад як studentUser:

```
mongosh -u studentUser -p student123 --authenticationDatabase students
```

Вивести всі документи з колекції users:

```
use students
```

```
db.users.find()
```

Перезапустити mongosh з автентифікацією, наприклад:

```
mongosh -u admin -p admin123 --authenticationDatabase admin
```

Виконання пункту 5

```
C:\Users\panda>mongosh -u studentUser -p student123 --authenticationDatabase students
Current Mongosh Log ID: 68b9951fc5e5a37fb1735188
Connecting to:  mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=students&appName=mongosh+2.5.7
Using MongoDB:      8.0.13
Using Mongosh:      2.5.7

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-09-04T13:10:45.011+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use students
switched to db students
students> db.users.find()
[
  {
    _id: ObjectId('68b979aa8da13c726e735189'),
    name: 'Іван Іванов',
    age: 20,
    group: 'CS-101',
    grades: [ 90, 85, 78 ]
  },
  {
    _id: ObjectId('68b979aa8da13c726e73518a'),
    name: 'Марія Петренко',
    age: 19,
    group: 'CS-101',
    grades: [ 80, 92, 81 ]
  },
  {
    _id: ObjectId('68b979aa8da13c726e73518b'),
    name: 'Олег Сидоренко',
    age: 21,
    group: 'CS-102',
    grades: [ 70, 75, 80 ]
  }
]
```

Рисунок 1.7– Перевірка доступу користувача studentUser до читання документів у колекції users.

```
students> exit

C:\Users\panda>mongosh -u admin -p admin123 --authenticationDatabase admin
Current Mongosh Log ID: 68b9967a7f2b8f8e7c735188
Connecting to:  mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.5.7
Using MongoDB:      8.0.13
Using Mongosh:      2.5.7

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-09-04T13:10:45.011+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> |
```

Рисунок 1.8– Перевірка доступу адміністратора admin

Обидва користувачі були успішно авторизовані, і їхні права доступу відповідають заданим ролям:

studentUser має доступ лише до бази students (читання/запис)

admin має глобальні права адміністрування

Це підтверджує, що автентифікація працює коректно, а ролі користувачів налаштовані відповідно до вимог лабораторної роботи.

Висновок

У ході виконання лабораторної роботи я ознайомився з основами роботи з документно-орієнтованою базою даних MongoDB. Було встановлено MongoDB Community Edition та MongoDB Shell, створено базу даних students з колекцією users, до якої додано документи, що містять інформацію про студентів.

Я навчився:

створювати бази даних і колекції;

додавати документи за допомогою команди insertMany;

створювати користувачів з різними рівнями доступу;

редагувати конфігураційний файл mongod.cfg для активації автентифікації;

підключатися до бази даних з авторизацією та перевіряти доступ до колекцій.

У результаті було успішно налаштовано адміністратора (admin) з глобальними правами та користувача (studentUser) з обмеженим доступом до бази students. Автентифікація працює коректно, що підтверджено тестовими підключеннями та виконанням запитів.

MongoDB продемонструвала гнучкість у роботі з даними, простоту адміністрування та ефективність у побудові сучасних веб-застосунків.

Відповіді на контрольні питання

1. Що таке MongoDB і в чому її особливість у порівнянні з реляційними СУБД?

MongoDB — це документно-орієнтована NoSQL база даних, яка зберігає дані у форматі BSON. Вона не потребує жорсткої схеми, дозволяє зберігати документи з різною структурою, підтримує масштабування, реплікацію та має потужні засоби фільтрації й агрегації. На відміну від реляційних СУБД, MongoDB не використовує таблиці, а працює з колекціями документів.

2. Яка команда в Mongo Shell використовується для створення нової бази даних?

Команда `use <назва_бази>` переключає на базу даних. Якщо база ще не існує, вона буде створена автоматично при першому записі.

3. Як додати кілька документів одночасно до колекції?

За допомогою команди `db.collection.insertMany([...])`, де в квадратних дужках передаються масив об'єктів-документів.

4. Як створити користувача з обмеженими правами доступу до конкретної бази даних?

У Mongo Shell потрібно виконати команду `db.createUser()` з роллю `readWrite` для потрібної бази, наприклад:

```
db.createUser({ user: "studentUser", pwd: "student123", roles: [ { role: "readWrite", db: "students" } ] })
```

5. Як перевірити всі документи у колекції?

Команда `db.collection.find()` виводить усі документи. Для зручного форматованого перегляду можна використати `db.collection.find().pretty()`.

6. Як перезапустити MongoDB сервер після внесення змін до конфігурації?

На Windows — вручну перезапустити `mongod.exe`, або через службу MongoDB у `services.msc`. На Linux — команда `sudo systemctl restart mongod`.

7. Як створити адміністратора в MongoDB?

У базі admin потрібно виконати команду `db.createUser()` з роллю `userAdminAnyDatabase`, наприклад:

```
db.createUser({ user: "admin", pwd: "admin123", roles: [ { role:
"userAdminAnyDatabase", db: "admin" } ] })
```