

DOI: <https://doi.org/10.36910/6775-2524-0560-2021-45-14>

УДК 004.6

**Христинець Наталія Анатоліївна**, к.т.н., старший викладач

<https://orcid.org/0000-0002-4836-7632>

**Лавренчук Світлана Василівна**, к.т.н., доцент

<https://orcid.org/0000-0002-5453-3924>

**Свиридюк Катерина Анатоліївна**, магістр

<https://orcid.org/0000-0003-0138-3931>

**Скригунець Василь Валерійович**, студент

Луцький національний технічний університет

## РОЗРОБКА МАСШТАБОВАНИХ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ REACT ТА БАЗИ ДАНИХ MONGODB

**Христинець Н.А., Лавренчук С.В., Свиридюк К.А., Скригунець В.В. Розробка масштабованих веб-додатків з використанням фреймворку React та бази даних MongoDB.** Преставлено особливості кластерної структури баз даних MongoDB з динамічною схемою документів для створення веб-додатку у вигляді інтернет-магазину на платформі Read. Запропоновано використання хмарної опції для безпеки та конфіденційності даних. Також розглянуто способи доступу до бази даних через систему запитів, API ключ та дашборд.

**Ключові слова:** веб-додаток, інтернет-магазин, база даних, Mongo DB, API

**Христинец Н.А., Лавренчук С.В., Свиридюк Е.А., Скрыгунец В.В. Разработка масштабируемых веб-приложений с использованием фреймворка React и базы данных MongoDB.** Представлены преимущества кластерной структуры баз данных MongoDB с динамической схемой документов для создания веб-приложения в виде интернет-магазина на платформе Read. Предлагается использование облачной опции для безопасности и конфиденциальности данных. Также рассмотрены способы доступа к базе данных через систему запросов, ключ API и дашборд.

**Ключевые слова:** веб-приложение, интернет-магазин, база данных, Mongo DB, API

**Khrystynets N., Lavrenchuk S., Svyrydiuk K., Skrygunets V. Development of scalable web applications using the React framework and the MongoDB database.** The advantages of the cluster structure of MongoDB databases with a dynamic document scheme for creating a web application in the form of an online store on the Read platform are presented. It is suggested to use cloud option for data security and privacy. Methods of accessing the database through the query system, API key and dashboard are also considered.

**Keywords:** web-application, online store, database, Mongo DB, API

**Постановка наукової проблеми.** Однією з найпопулярніших сучасних технологій у сфері веб-розробок є MEAN – це абревіатура, до якої входять технології MongoDB, ExpressJs, Angular і NodeJs. Фактично це набір комплексного програмного забезпечення, який використовують для веб-розробок.

Зазвичай, коли постає необхідність вибирати базу даних для веб-додатку, розробники витрачають багато часу на порівняння кількох операційних баз даних, щоб вибрати ту, яка найкраще відповідає їхнім робочим навантаженням, вимогам, характеристикам. В одних випадках, якщо проектується сайт-візитка, чи промосайт, то потреби щодо баз даних можуть включати спрощене моделювання даних і виконання елементарних транзакцій, в інших випадках, зокрема в проектуванні інтернет-магазину, важливими характеристиками стають ще й продуктивність читання/запису, горизонтальне масштабування та відмовостійкість. Традиційно, початкове звернення уваги починається з категорій SQL або NoSQL. Популярним сучасним вибором з класу баз даних NoSQL є MongoDB [1]. База чудово працює для багатьох сучасних програм, таких як мобільні, ігрові та інтернет-додатки завдяки високій продуктивності та широкими функціональним можливостям, які забезпечують максимальну зручність її використання. Питання методів доступу до MongoDB, яка використовується при розробці інтернет-магазину у якості серверного кластера для розміщення основних елементів веб-ресурсу, є сьогодні надзвичайно актуальною задачею в плані безпеки та конфіденційності даних.

**Аналіз досліджень.** MongoDB пропонує як локальні, так і хмарні варіанти розгортання: Для локальних розгортань MongoDB пропонує як спільноту, так і корпоративну версію бази даних. Спільнота MongoDB є доступним і безкоштовним джерелом видання MongoDB. MongoDB Enterprise доступна як частина підписки MongoDB Enterprise Advanced і включає комплексну підтримку розгортання MongoDB. MongoDB Enterprise також додає такі корпоративні функції, як підтримка LDAP та Kerberos, шифрування на диску та аудит. MongoDB Atlas - це розміщена в хмарі опція

служби MongoDB Enterprise, яка не вимагає додаткових витрат на встановлення та пропонує безкоштовний рівень для початку роботи [2].

Завдяки вбудованій підтримці пошуку, аналітики та периферійної підтримки, доступній за допомогою спільного API запитів і побудованої на основі моделей даних, які люблять розробники, MongoDB робить роботу з даними легкою – для будь-якого випадку.

MongoDB побудована на масштабованій архітектурі, яка стала популярною серед розробників масштабованих додатків з еволюційними схемами даних. Як база даних документів, MongoDB спрощує розробникам зберігання структурованих або неструктурованих даних. Він використовує формат, подібний до JSON, для зберігання документів. База даних представлена у вигляді фізичного сховища колекцій. Кожна БД має свій власний набір файлів в файловій системі. Зазвичай, один MongoDB сервер має кілька БД. Досліджено [3], що протягом 2021 року трійка лідерів практично не змінилась: MS SQL та PostgreSQL набагато популярніші серед прихильників SQL, тоді як SQLite і MongoDB більш популярні серед тих, хто лише зрідка використовує SQL (рис. 1). Велика кількість розробників використовують дві або більше баз даних одночасно. Так, користувачі MongoDB також використовують утиліту MySQL Workbench.

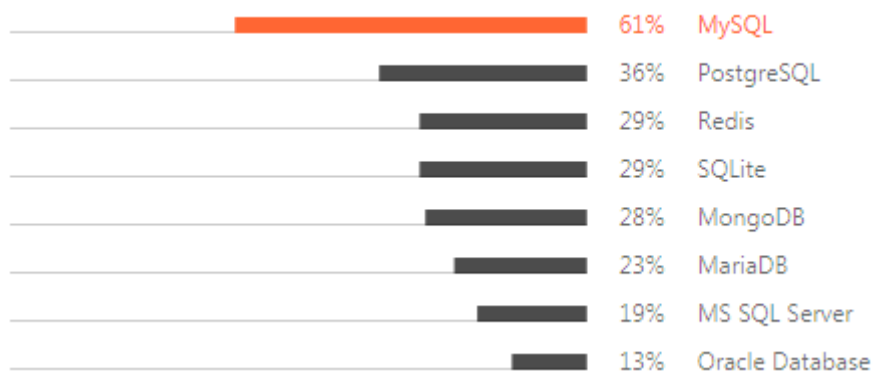


Рис. 1. Рейтинг баз даних за 2021 рік (версія Devecosystem)

#### **Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.**

Інтернет-магазин розроблений на базі фреймворку React, який є популярним інструментом для створення інтерактивних програм. Використовуючи бібліотеку, можна розгорнути нові проекти React на будь-якому постачальнику статичного хостингу. Це чудово для продуктивності та безпеки, але також має обмеження: вміст часто має бути жорстко закодованим у програмі. Основний інструментарій створення веб-додатку – ReactJS. Це інтерфейсний фреймворк, який не взаємодіє з базою даних, якщо не створено REST API із серверною частиною. Щоб редагувати та оновлювати текст або зображення, потрібно внести зміни в код і повторно розгорнути всю програму. React дуже добре обробляє дані API. Щоб зробити вміст доступним для редагування, слід використовувати систему керування вмістом на основі API.

Процес створення дизайн-схеми (рис. 2) MongoDB під час роботи з обліковим записом одного користувача та кількома користувачами (обліковий запис організації з ролями) відбувався наступним чином: в обліковому записі компанії є адміністратор (власник) та інший запрошений користувач, і кожному користувачеві призначається рівень дозволу, або рівень доступу.

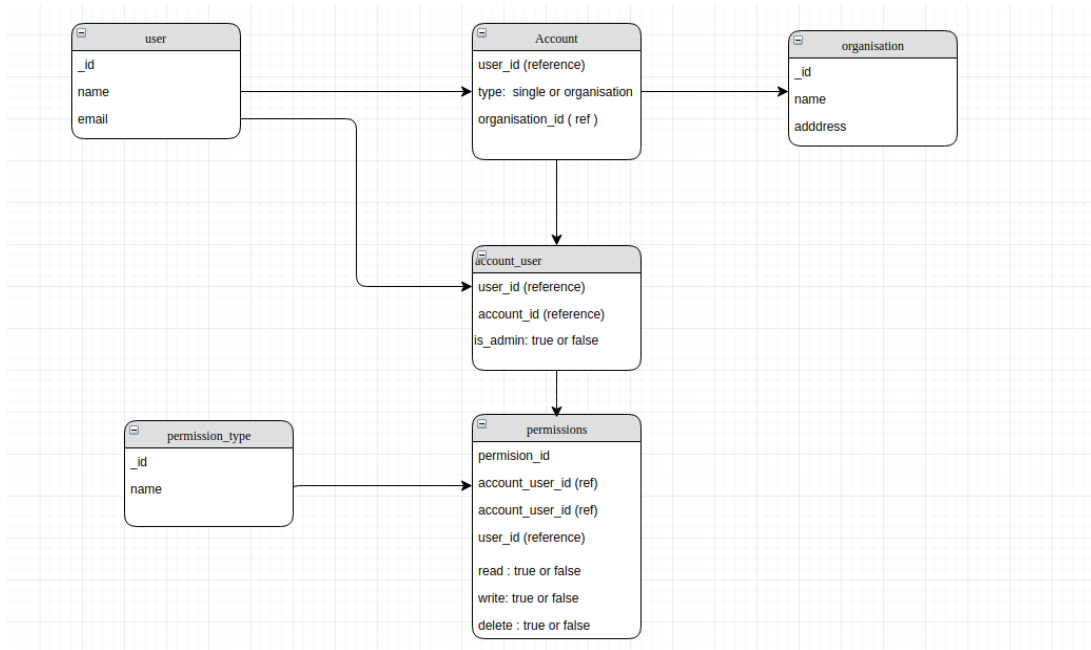


Рис. 2. Схема взаємопов'язаних таблиць бази даних інтернет-магазину

Один користувач завжди буде пов'язаний лише з одним обліковим записом, тобто його не можна знову запросити в інший обліковий запис, або розпочати новий обліковий запис, якщо він уже є частиною наявного облікового запису. Адреса виставлення рахунку та доставки специфічна для облікового запису, а не для користувача у випадку облікового запису компанії. Адреса переходу користувача на обліковий запис компанії буде адресою облікового запису компанії.

MongoDB є документо-орієнтованою базою. Документ – це набір пар «ключ - значення», він має динамічну схему. Це означає, що документ в одній і тій же колекції не зобов'язаний мати один однаковий набір полів або структуру, а загальні поля в колекції можуть мати різні типи даних.

Одним із варіантів підключення бази даних є firebase. Ця хмарна технологія звільняє розробників від багатьох питань користувацького досвіду, при її використанні не потрібно керувати серверами, не потрібно писати API. Firebase – це сервер, API і сховище даних водночас з налаштуваннями, які можна змінити відповідно до більшості потреб користувача. Інколи при роботі з Firebase доведеться використовувати інші частини Google Cloud для якихось розширених програм. При залученні MongoDB як firebase, MongoDB надає можливість створити кластер на їхньому сервері (рис. 3).

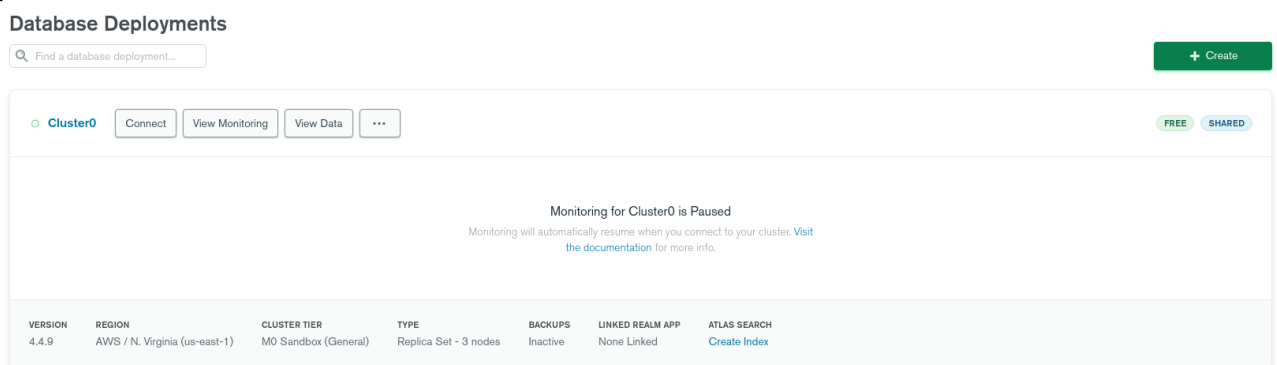


Рис. 3. Елемент кластеру бази даних MongoDB

Було розроблено частину авторизації, де користувачеві потрібно обмежити доступ до ресурсу, перевіряючи дозвіл, який він має. Було розроблено проміжне програмне забезпечення для авторизації, яке запускається після програмного забезпечення для автентифікації, яке перевіряє наявність дозволів на доступ. Проте, коли необхідно було отримати доступ до даних облікового запису на основі поточного користувача, виникла проблема, оскільки довелося шукати документ на основі посилання на `objectId`. При продовженні роботи з поточним дизайном дана ситуація була передбачувана, але пошук у документі за допомогою посилання `objectId`, здається, не дуже вдала ідея. Тому, було використано проміжне програмне забезпечення для авторизації.

Доступ до бази даних відбувається через API ключ, оскільки за допомогою API можна створювати кластери та автоматизувати їх обслуговування у базі даних сайту. Цей ключ MongoDB генерує з даних користувача. В нашому випадку використання MongoDB, як firebase, ми вибрали «Connect your application» (рис. 4), після чого було видано APU url на кластер бази даних MongoDB.

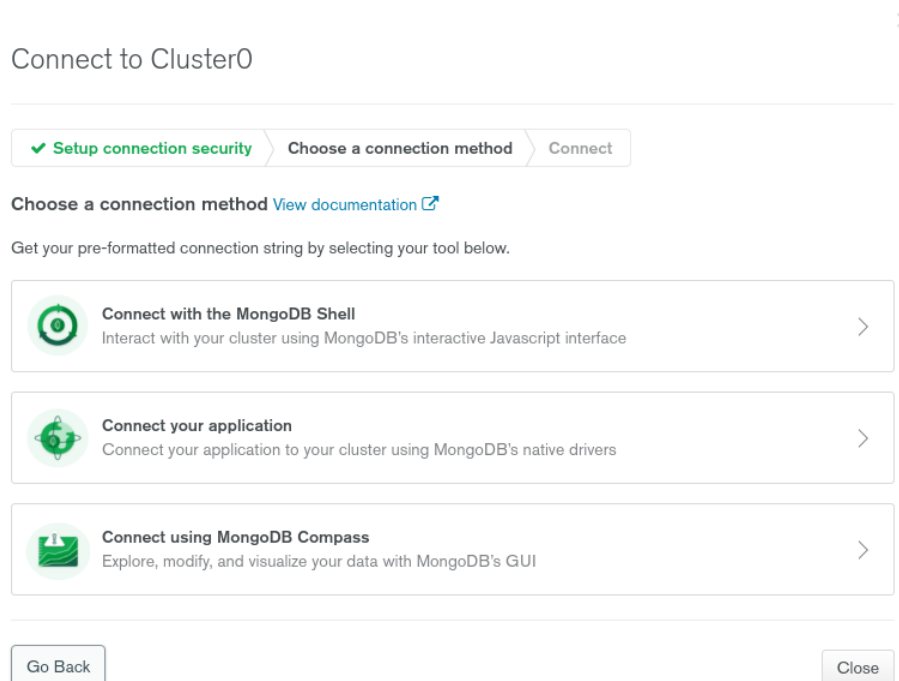


Рис. 4. Вікно налаштувань кластерів служби MongoDB

Підключення відбувалось за допомогою запиту, за допомогою функції «connect», яка описана в пакеті «mongoose».

```
await mongoose.connect(DB_URL,{
  useNewUrlParser: true,
  useUnifiedTopology: true
})
```

Наступним етапом відбувалось створення моделі, після імпортування mongoose. Для створення моделі була необхідна «Schema», і загалом, модель користувача має наступний вигляд:

```
const {Schema, model} = require('mongoose');

const UserSchema = new Schema({
  email: {type: String, unique: true, required: true},
  password: {type: String, required: true},
  isActivated: {type: Boolean, default: false},
  isAdmin: {type: Boolean, default: false},
  activationLink: {type: String}
});
```

```
module.exports = model('User', UserSchema);
```

Звертання до бази даних відбувалося за допомогою моделі, таким чином:

```
const user = await UserModel.create({email, password: hashPassword, activationLink})
```

Результат виконаної операції відображено в кластері на UI в MongoDB (рис. 5).



Рис. 5. Фрагмент відображення звертання до MongoDB за допомогою моделі

Зміни параметрів аутентифікації можна здійснювати в дашборді (інформаційній панелі) MongoDB (рис. 6). Інформаційні панелі являють собою набір діаграм, зібраних для створення єдиного відображення даних.

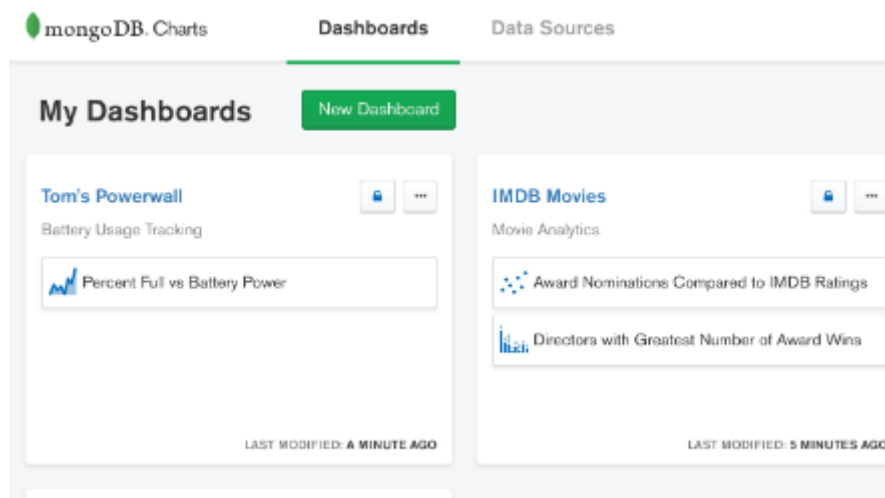


Рис. 6. Інформаційна панель діаграм MongoDB

Кожна діаграма показує дані з однієї колекції або подання MongoDB, тому інформаційні панелі мають важливе значення для того, щоб отримати уявлення про декілька координаційних центрів даних на одному дисплеї. Інформаційними панелями можна поділитися з іншими користувачами. На сторінці інформаційної панелі відображаються всі інформаційні панелі, до яких є доступ. Параметри MongoDB можна відстежувати за допомогою п'яти ключових категорій показників в дашборд: пропускну здатність, насиченість ресурсів, використання ресурсів, помилки і продуктивність. Однак, доступні показники та їх відносна важливість залежать від механізму зберігання, який використовується. У сучасних версіях MongoDB це WiredTiger, який зараз є стандартним, або MMARv1, який використовувався за замовчуванням у попередніх версіях MongoDB. Наприклад, адміністратори користувачів MongoDB Charts мають право додавати користувачів, видаляти користувачів, редагувати профілі інших користувачів та змінювати їхні паролі. Адміністратори користувачів можуть отримати доступ до перегляду керування користувачами, натиснувши вкладку «Керування користувачами» на верхній навігаційній панелі.

#### **Висновки та перспективи подальшого дослідження.**

При створенні веб-додатків сучасним фахівцям не обійтись без фреймворків та бібліотек JavaScript, часто буває досить важко вибирати між популярними фреймворками, бо у кожного є переваги та недоліки. Нами був обраний фреймворк React для побудови інтернет-магазину, який дозволив реалізувати усі необхідні параметри веб-ресурсу, такі як картки товарів, корзина, категорії та запити, а також аутентифікацію користувачів. Для реалізації була використана документо-орієнтована база MongoDB, яка дозволяє оперувати документами, що зберігаються в колекціях, які є аналогом звичних SQL-таблиць. Для роботи з документами було передбачено операції пошуку, вставки, видалення і оновлення. Для пошуку документів в колекції використано метод запитів за зразком, підтримку сортування, проєкції, перегляду результатів запиту. Масштабованість в MongoDB досягається за рахунок поділу документів з колекції по вузлах на підставі обраного ключа (shard key).

Доступ до бази даних реалізовано методами API ключів, інформаційної панелі та firebase. Подальші дослідження можуть бути спрямовані на вдосконалення методів захисту баз даних, цілісності даних в процесі їх модифікації тощо.

#### References

1. Yoon J. A method and tool to recover data deleted from a MongoDB / J. Yoon, S. Lee. // Digital Investigation. – 2017. – №24. – С. 106–120.
2. Rokatsky M. 5 Best Tools for MongoDB Atlas in 2021 [Online] / Max Rokatsky // OTUS. – 2021. – Resource access mode: <https://habr.com/ru/company/otus/blog/585568/>.
3. Devecosystem 2021. Databases [Електронний ресурс] // Jet Brains. – 2021. – Режим доступу до ресурсу: <https://www.jetbrains.com/ru/devecosystem-2021/databases/>.
4. Tsukasa K. A Proposal of Transaction Processing Method for MongoDB / K. Tsukasa, I. Masahiko, S. Kenji. // Procedia Computer Science. – №96. – С. 801–810.