



Технически Университет - София



Склад с MongoDB

курсова работа по
“Нерелационни бази с данни”

Имена на студент: **Кристиян Стойков**

Факултетен номер: **121221086**

Група: **45**

Специалност: **компютърно и софтуерно инженерство**

Образователно-квалификационна степен: **бакалавър**

Съдържание

1.	Въведение	3
	Фигура 1: Схема на примерен запис в MongoDB	Error! Bookmark not defined.
2.	Теоретична и технологична обосновка	4
3.	Архитектура и реализация	5
4.	Използвана литература (библиография)	7
	4.1 Използвани съкращения	7

1. Въведение

Проектът представлява уеб приложение за управление на продукти, поръчки и клиенти. Основната цел е да се разработи система за съхраняване и обработка на информация за клиенти, продукти и поръчки. Приложението е предназначено за използване от администратори и крайни потребители, като осигурява различни нива на достъп до данните според роля. За всеки потребител се създава сесия, която автоматично се изтрива от базата чрез TTL индекс.

Приложението включва административен панел, който позволява на администраторите да управляват потребители, да добавят продукти и да наблюдават поръчките на клиенти в реално време. Редовните потребители могат да влязат, да преглеждат продуктите.

Целта на този проект е да се изгради проста, но функционална система за управление на склад/поръчки. MongoDB е избрана за база данни поради нейната гъвкавост с динамични структури от данни, лекота на интегриране с Next.js и вградена поддръжка за функции като TTL индекси, които помагат за ефективното управление на изтичането на сесията без допълнителна логика.

- Централизирано съхранение на данни за клиенти, продукти и поръчки
- Управление на достъпа чрез ролеви модел (администратор и потребител) и управление на сесията
- Изпълнение на CRUD операции върху данните
- Повишаване на проследимостта и отчетността

2. Теоретична и технологична обосновка

Нерелационни бази данни: Това са бази данни, които не използват традиционната таблица със схеми и релации. Те са създадени за гъвкаво съхранение на данни и поддръжка на различни структури: документи, ключ-стойност, графи и колони. Основни им предимства са следните:

- Липса на фиксирана схема
- Лесна хоризонтална мащабируемост
- Висока производителност при големи обеми данни

В този проект е използвана е документно-ориентирана база – MongoDB, която съхранява данните във формат BSON/JSON. Това позволява лесна сериализация и директна интеграция с JavaScript-базиран back-end.

Технологиите, които са използвани в проекта са:

- MongoDB – за съхранение на данни
- nextJS – JavaScript среда за уеб базирани приложения
- crypto – JavaScript библиотека за хеширане на текст и др.

Линк към проекта в github: <https://github.com/kristiyanstoykov/sem8-nbd-project/tree/main>

3. Архитектура и реализация

Структурата в базата от данни е:

- Потребител:

```
{
  "_id": ObjectId,
  "name": "steveO",
  "email": "email@email.email",
  "password": "some_hash",
  "role": "admin"|"user"
}
```
- Продукти:

```
{
  "_id": ObjectId(),
  "name": "some_name"
  "price": 500,
  "quantity": 3
}
```
- Поръчки:

```
{
  "_id": ObjectId,
  "items": Array,
  "status": "completed|pending|cancelled|processing"
  "client_id": "generated_id",
  "total_price": 2350,
  "createdAt": "2025-03-29T13:00:00.000Z",
  "updatedAt": "2025-03-29T13:10:00.000Z",
}
```
- Сесия:

```
{
  "_id": ObjectId,
  "userId": some_id,
  "role": "admin"|"user",
  "expiresAt": "2025-03-29T13:00:00.000Z", (index)
  "createdAt": "2025-03-29T13:00:00.000Z"
}
```

Основните операции, които ще се извършват с данните:

- Create: (admin) Добавяне на нови потребители и продукти
- Read: Списъци с продукти, (admin) списък с поръчки, клиенти и продукти
- Update: (admin) Редактиране на съществуващ продукти и потребители
- Delete: (admin) Изтриване на продукти и потребители
- Филтриране на данните по различни критерии и сортиране.

- Authentication: Вход с потребителско име и парола със запазване на сесията в базата и изтриване при изминато определено време:
 - Сесията се записва в MongoDB колекция sessions с поле expiresAt.
 - Сесията се съхранява в колекцията sessions в MongoDB заедно с поле expiresAt, което показва кога трябва да изтече сесията. TTL (Time To Live) индекс се създава в това поле, което казва на MongoDB автоматично да изтрие записа на сесията, след като времето изтече. Това гарантира, че изтеклите сесии се прекратяват автоматично от базата данни, без да е необходимо ръчно изтриване.
- RBAC: Ограничен достъп до функции спрямо роля
- Филтриране на поръчки и взимане на името на клиента вързан към поръчката с една заявка. Прави се \$lookup, за да се съединят поръчките със съответния потребител от колекцията users (съвпадение по client_id). Резултатът е опростен с \$project, за да включва само специфични полета като информация за поръчка и потребителско име (от таблицата users) и накрая сортиран се сортира по _id на поръчка.

```

if (status) pipeline.push({ $match: { status } });
pipeline.push(
{
  $lookup: {
    from: "users",
    let: { searchId: { $toObjectId: "$client_id" } },
    pipeline: [
      { $match: { $expr: { $eq: ["$_id", "$$searchId"] } } },
      { $project: { name: 1 } }
    ],
    as: "user"
  },
  { $unwind: "$user" },
  {
    $project: {
      _id: 1,
      createdAt: 1,
      updatedAt: 1,
      status: 1,
      total_price: 1,
      user_name: "$user.name"
    }
  },
  { $sort: { _id: 1 } }
);

```

4. Използвана литература (библиография)

[1] MongoDB Documentation: <https://www.mongodb.com/docs/>

[2] next.js Documentation: <https://nextjs.org/docs>

[4] Web Dev Simplified authentication tutorial:
https://www.youtube.com/watch?v=yoiBv0K6_1U&list=PL5d-f9QMU2-cmXgL5O1wvLiJGyQHwjzr&index=3&t=2898s&pp=gAQBiAQB

4.1 Използвани съкращения

1. НБД – Нерелационна база данни
2. RBAC – Role-Based Access Control
3. TTL – Time To Live