**Звіт**

**Лабораторна робота № 4**

**З дисципліни «Бази даних та інформаційні системи»**

**Виконував: Чорноконь Денис Сергійович**

**Тема:** розробка Інформаційної Системи (ІС) для обробки та візуалізації даних з використанням MEAN стеку.

**Мета:** розробка повнофункціональної інформаційної системи для обробки та візуалізації даних в обраній області за допомогою MEAN стеку (MongoDB, Express.js, React, Node.js).

Система повинна включати роботу з базою даних MongoDB,

серверний API на основі Express.js та Node.js,

а також клієнтську частину, реалізовану за допомогою React.

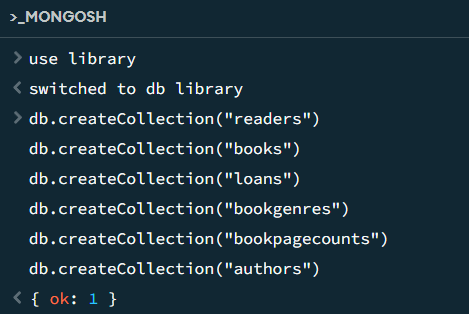
**Хід роботи**

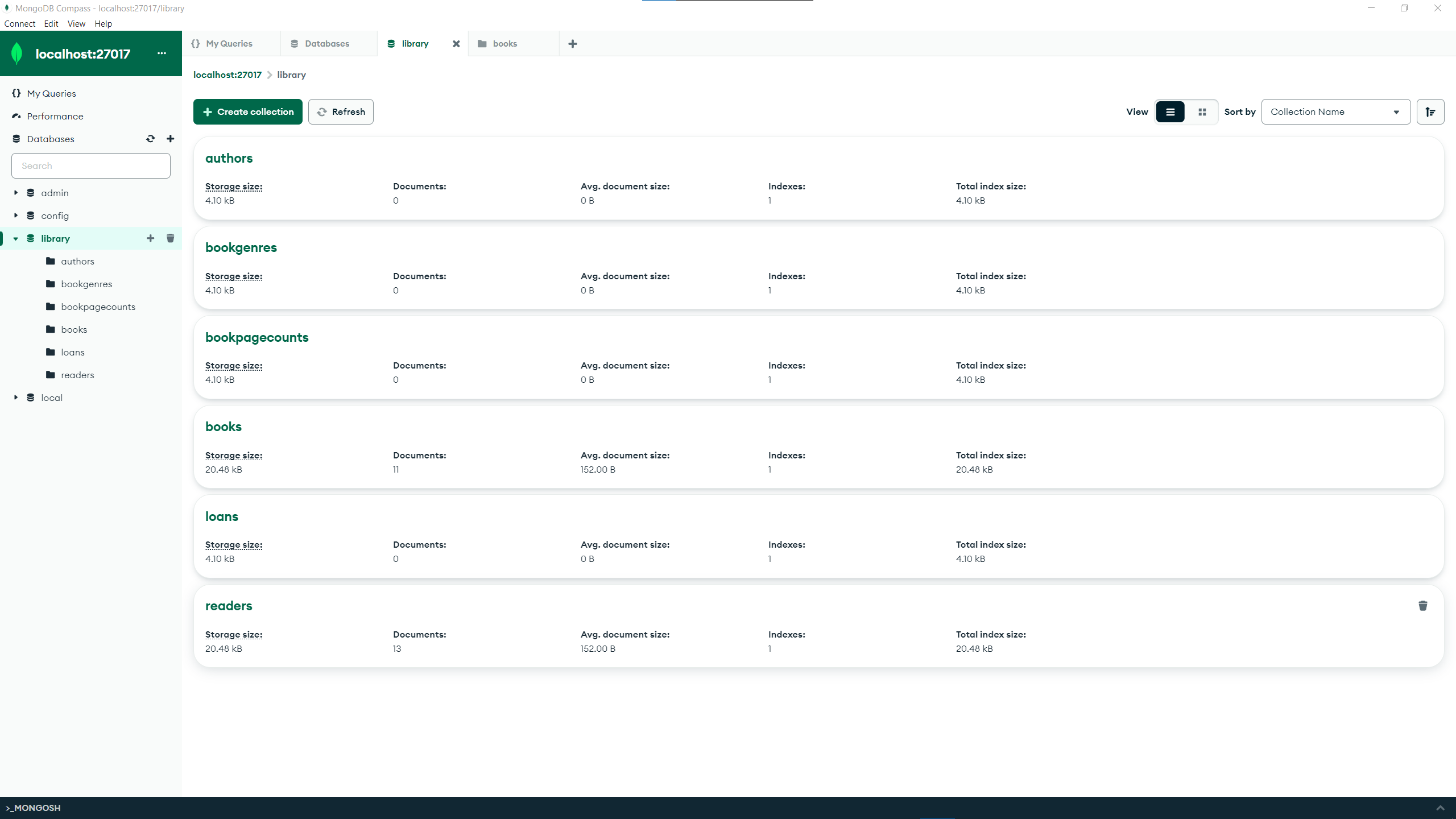
1. Проектування ІС:

- Вибір області застосування та визначення функціональних вимог до системи.

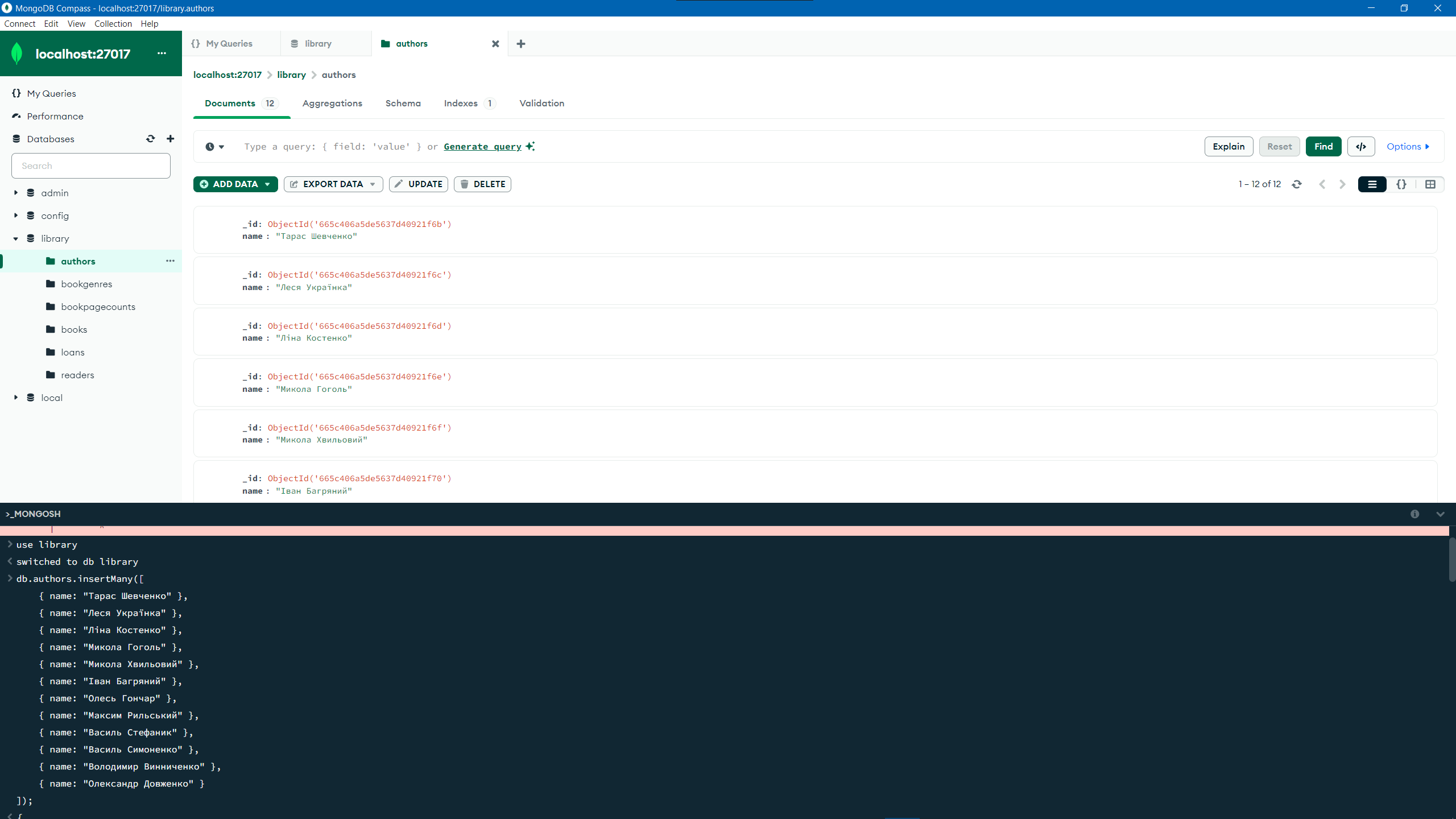
- Проектування структури бази даних в MongoDB.

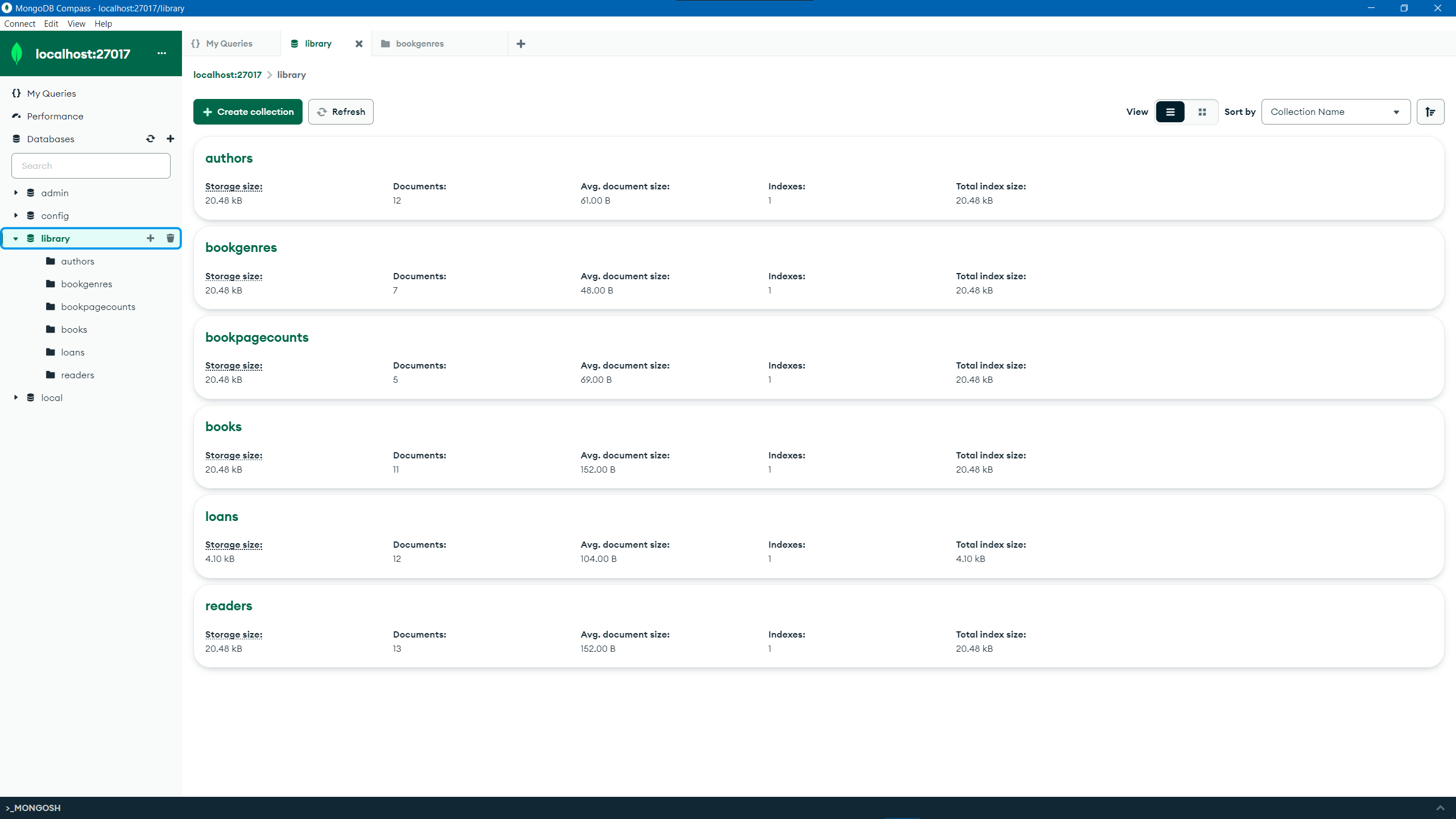
MongoDB встановлено, створено базу даних бібліотеки. В БД додано колекції.





Колекції заповнено даними відповідно до БД бібліотеки за допомогою методу insertMany().

В розділі Documents кожної колекції можна побачити кількість доданих елементів.



2. Розробка серверної частини:

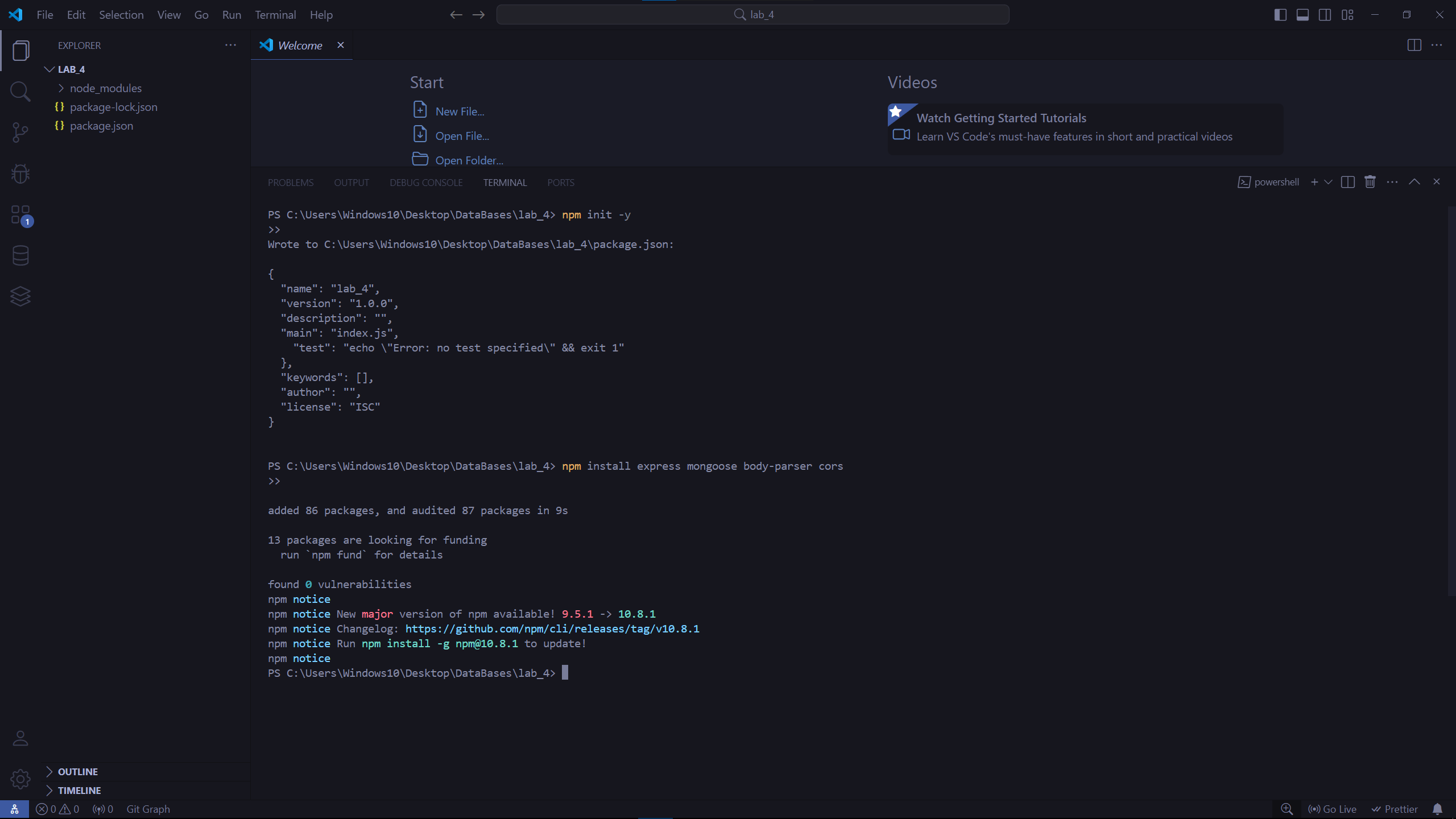
- Налаштування бази даних MongoDB.

- Реалізація API за допомогою Express.js та Node.js для доступу та управління даними (CRUD операції).

Дану частину буде поділено на три етапи, спочатку налаштування проекту, потім підключення до MongoDB, і реалізація CRUD операцій.

1. Налаштування проекту.

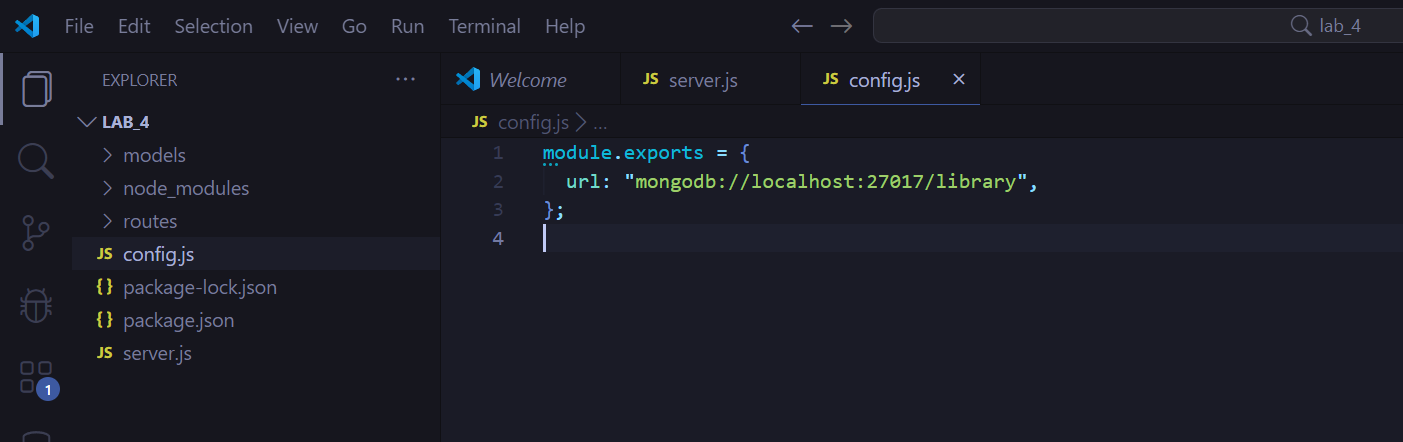
Створюємо новий проект Node.js, та встановлюємо необхідні залежності.



Також створюємо файл server.js для налаштування підключення до MongoDB.

1. Підключення до MongoDB.

Створюємо файл конфігурації для підключення до MongoDB – config.js.

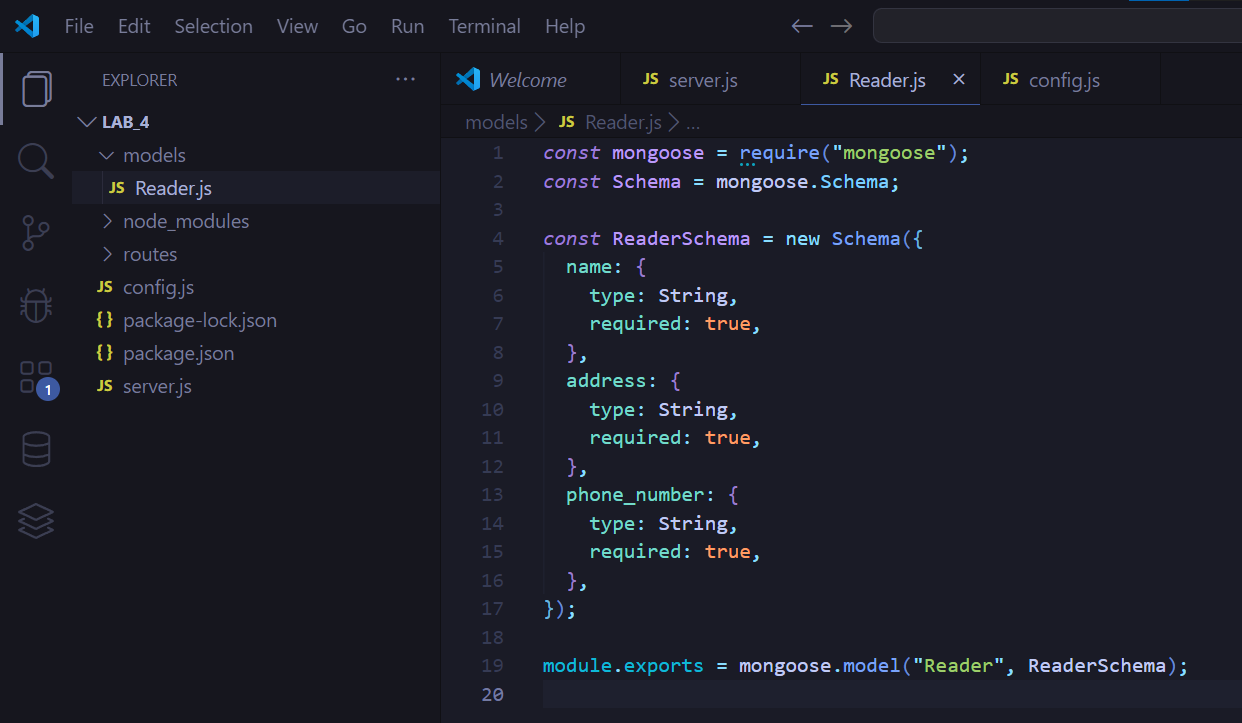


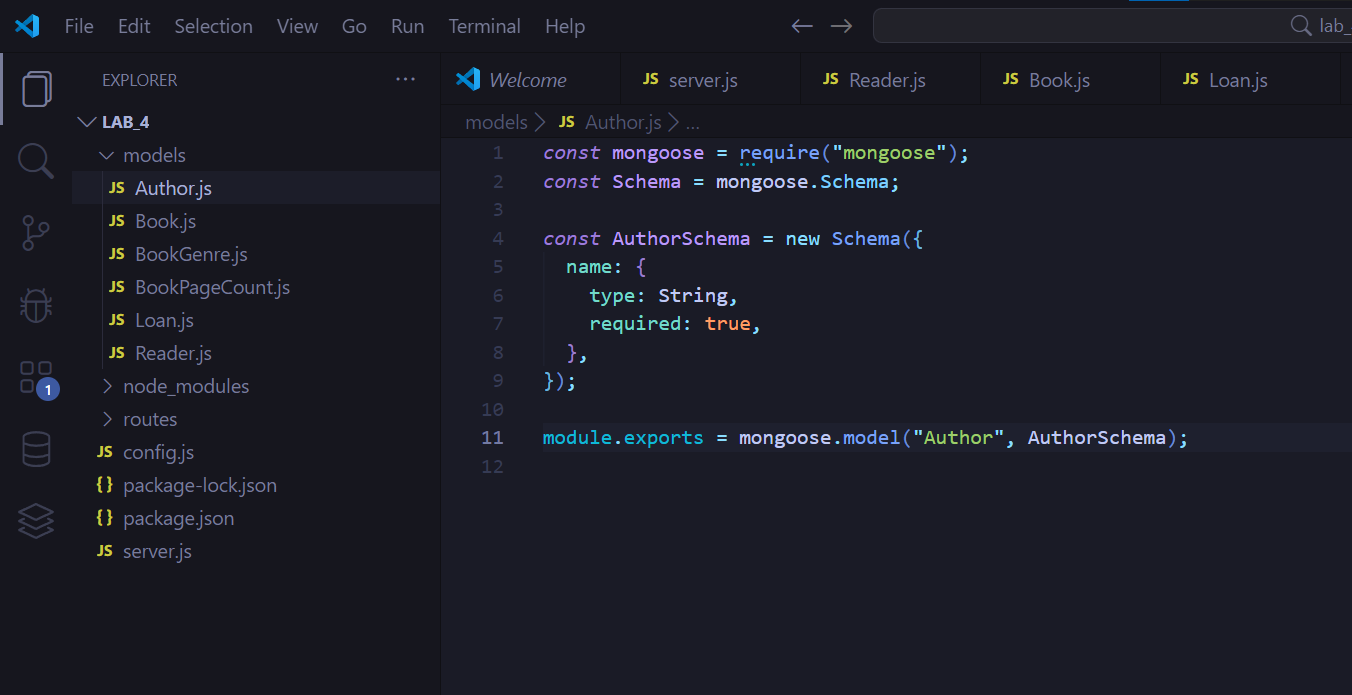
Налаштовуємо підключення у файлі server.js.



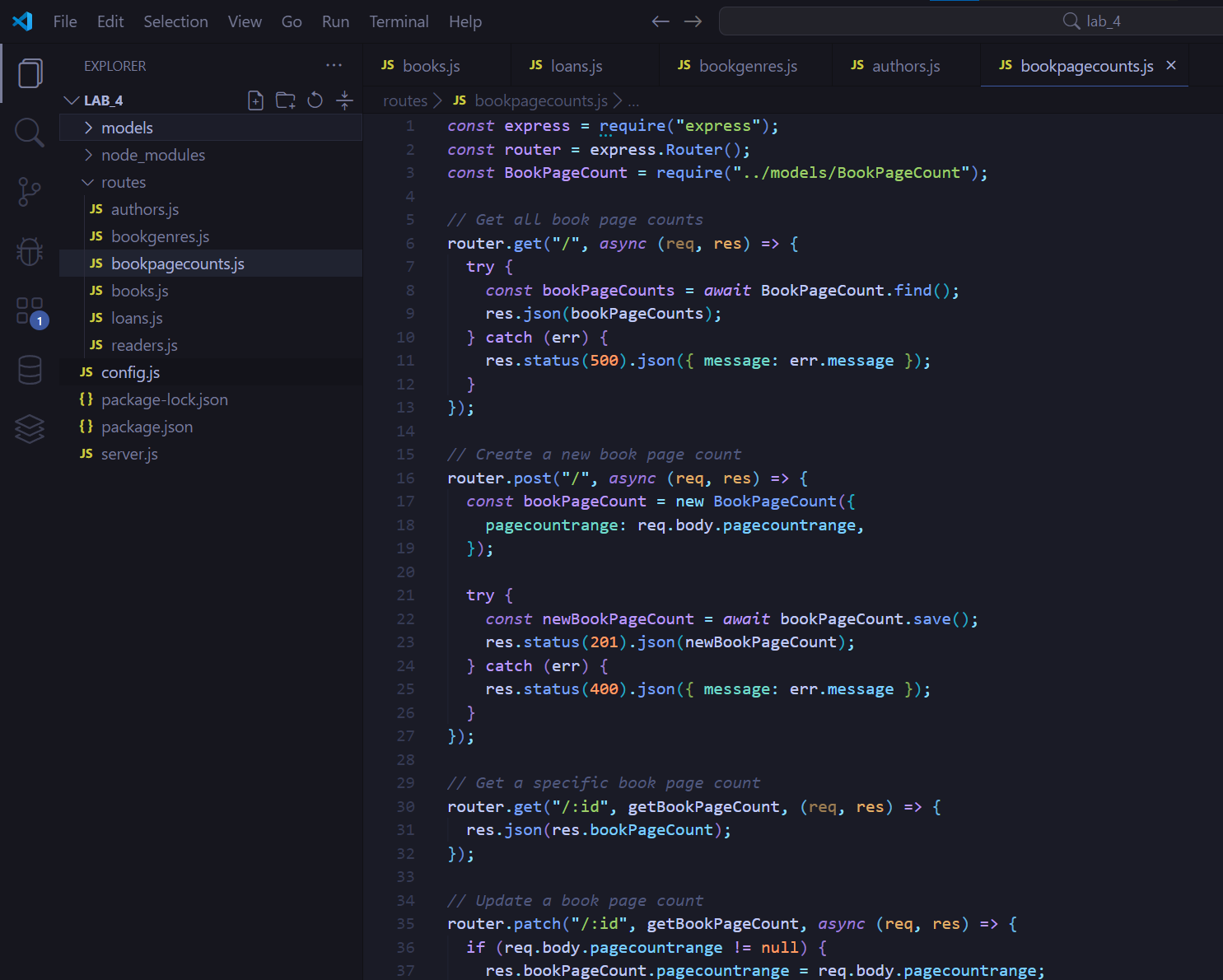
1. Реалізація CRUD операцій

У директорії models створюємо моделі для MongoDB.





У директорії routes створюємо файли маршрутів.



Кожен файл у директорії routes відповідає за CRUD операції для певної колекції (операції взяті з попередньої лабораторної). Усі маршрути підключені у головному файлі server.js.

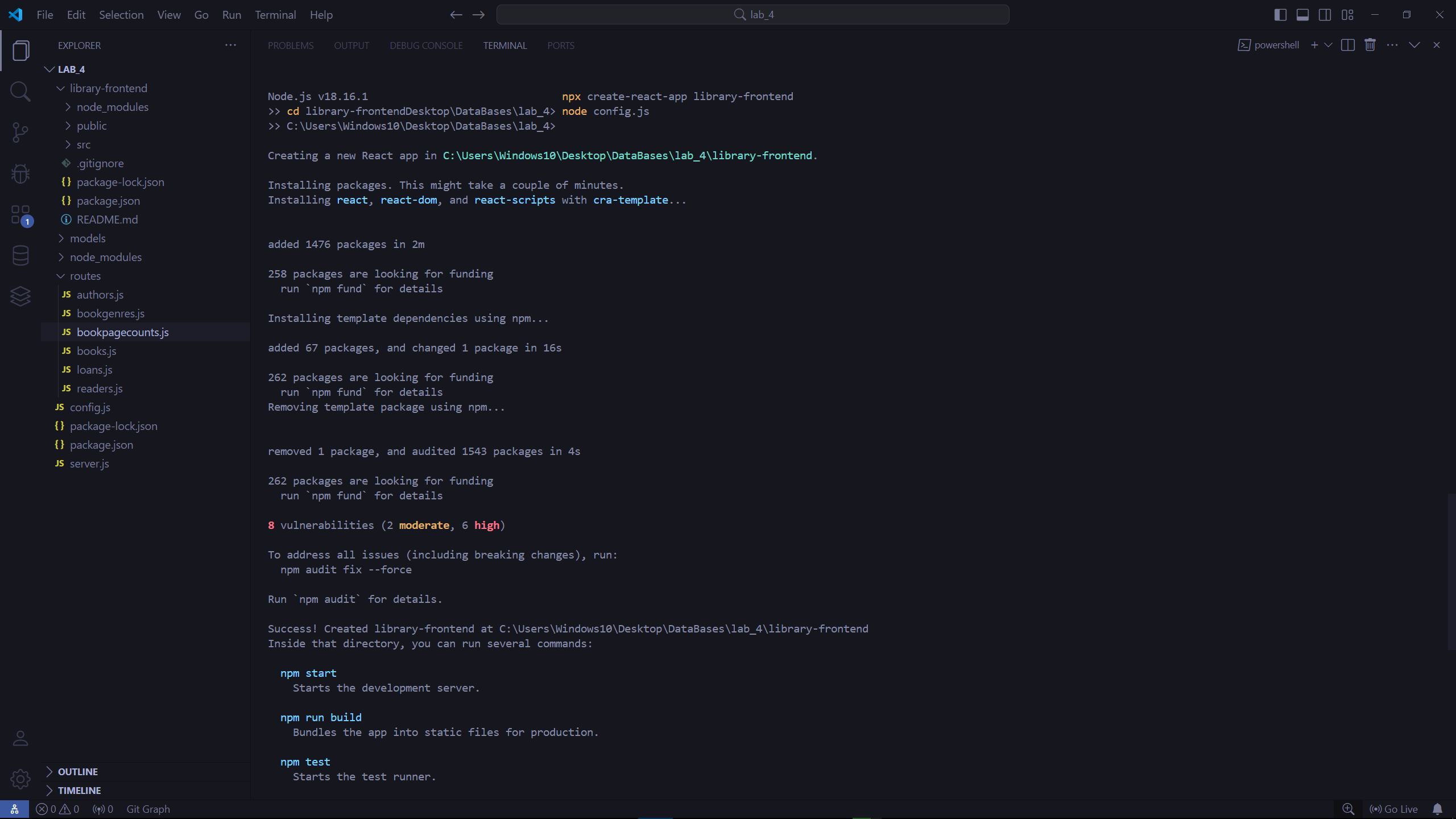
3. Розробка клієнтської частини:

- Розробка користувальницького інтерфейсу з використанням React.

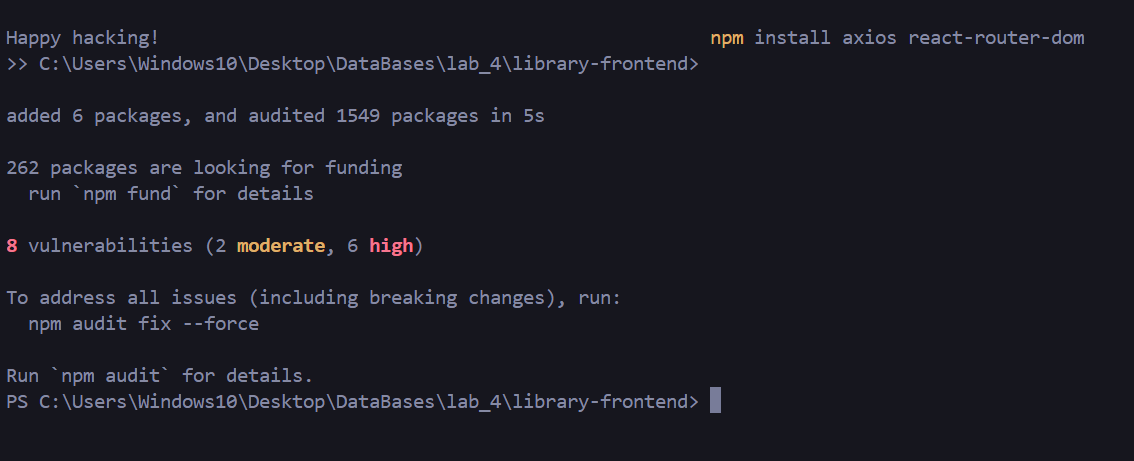
- Реалізація функцій пошуку, сортування та фільтрації даних.

- Візуалізація даних за допомогою графіків, діаграм та інших засобів візуалізації.

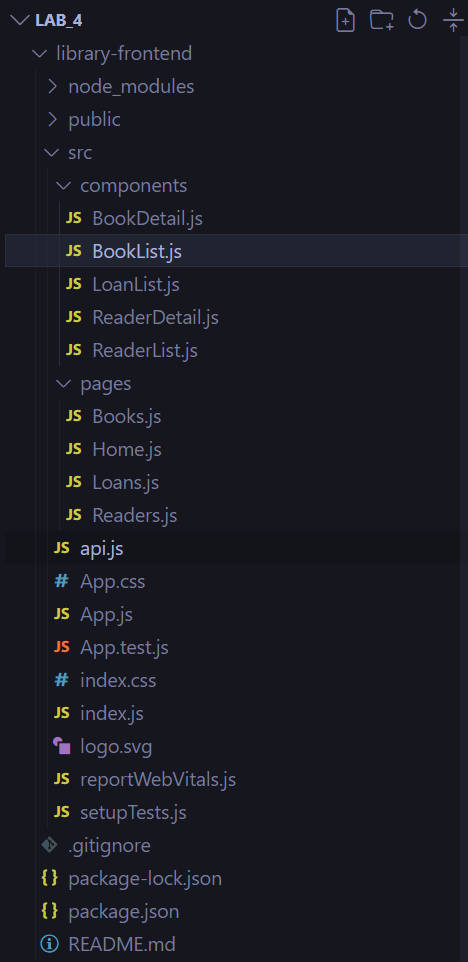
Спочатку створюємо React проект (в директорії library-frontend).



Встановлюємо пакет Axios для виконання HTTP запитів та React Router для маршрутизації.

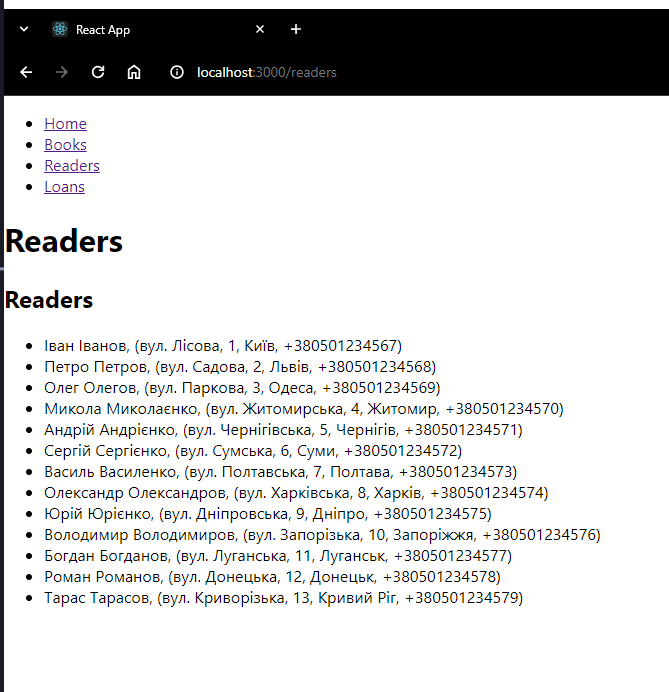


Створимо структуру директорій.



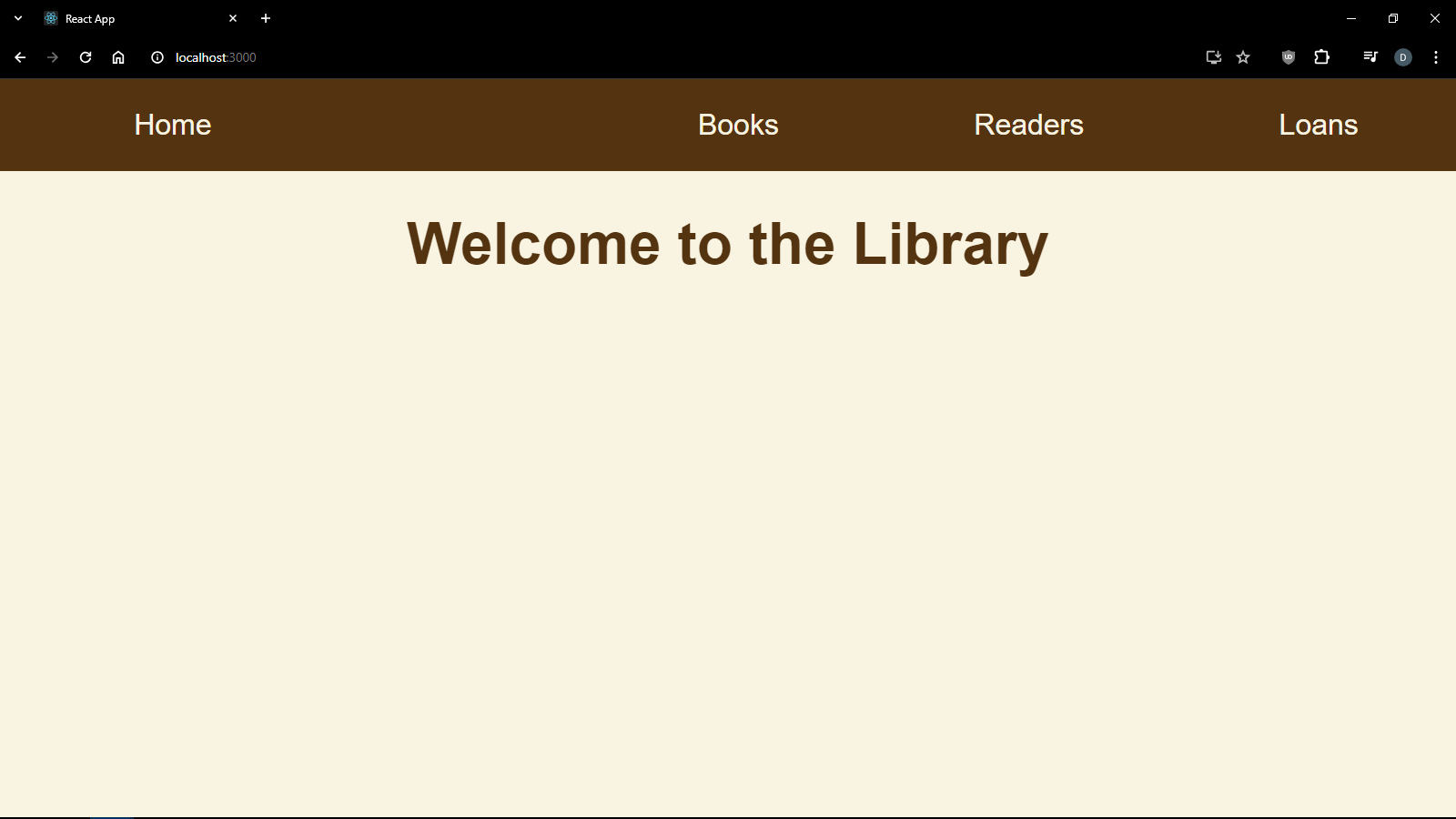
Починаємо роботу над Front-End, заповнюючи компоненти.

На даному етапі сторінки виглядають так:

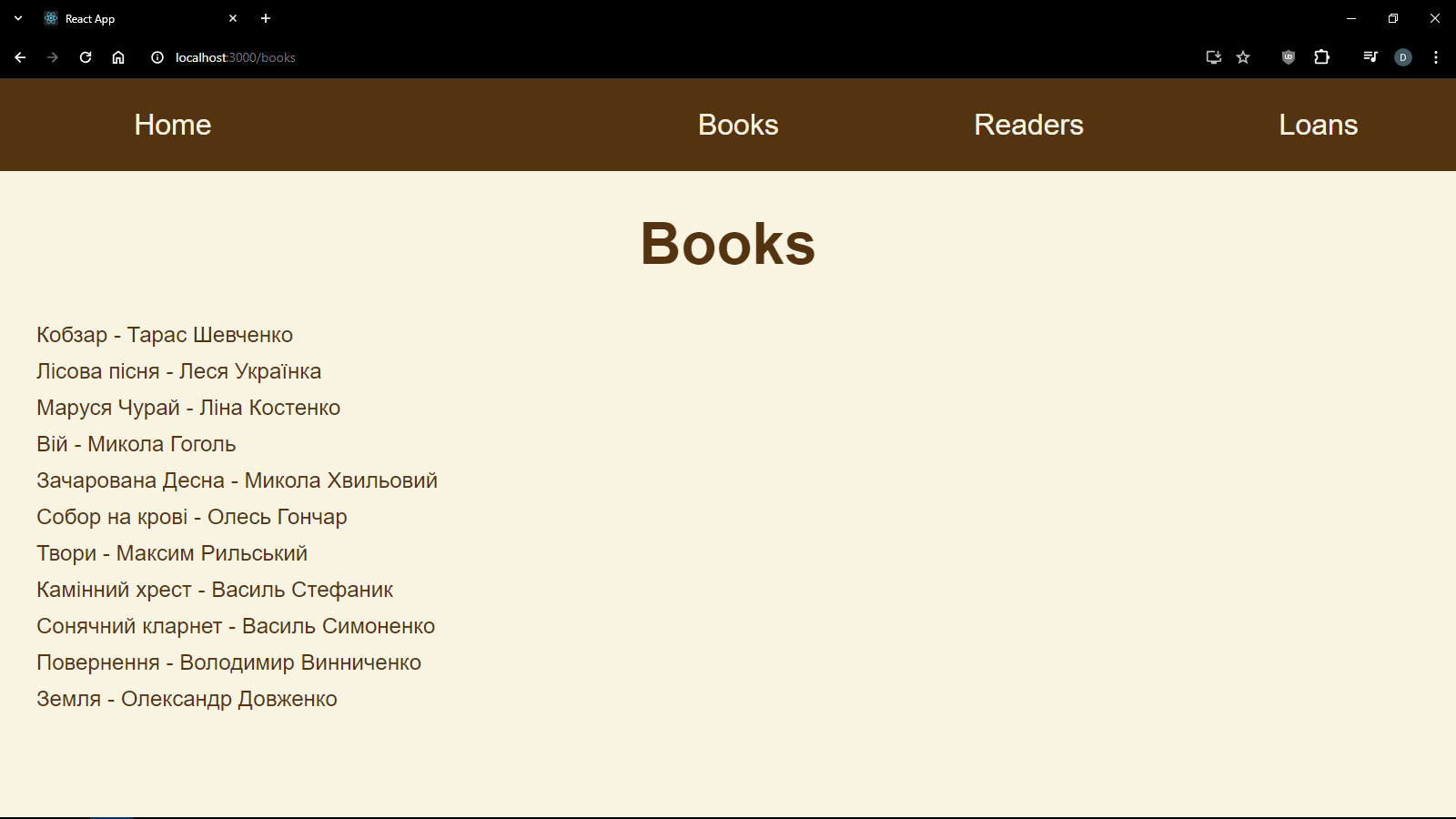


Трохи стилізуємо сторінки за допомогою css. На даному етапі є 4 сторінки:

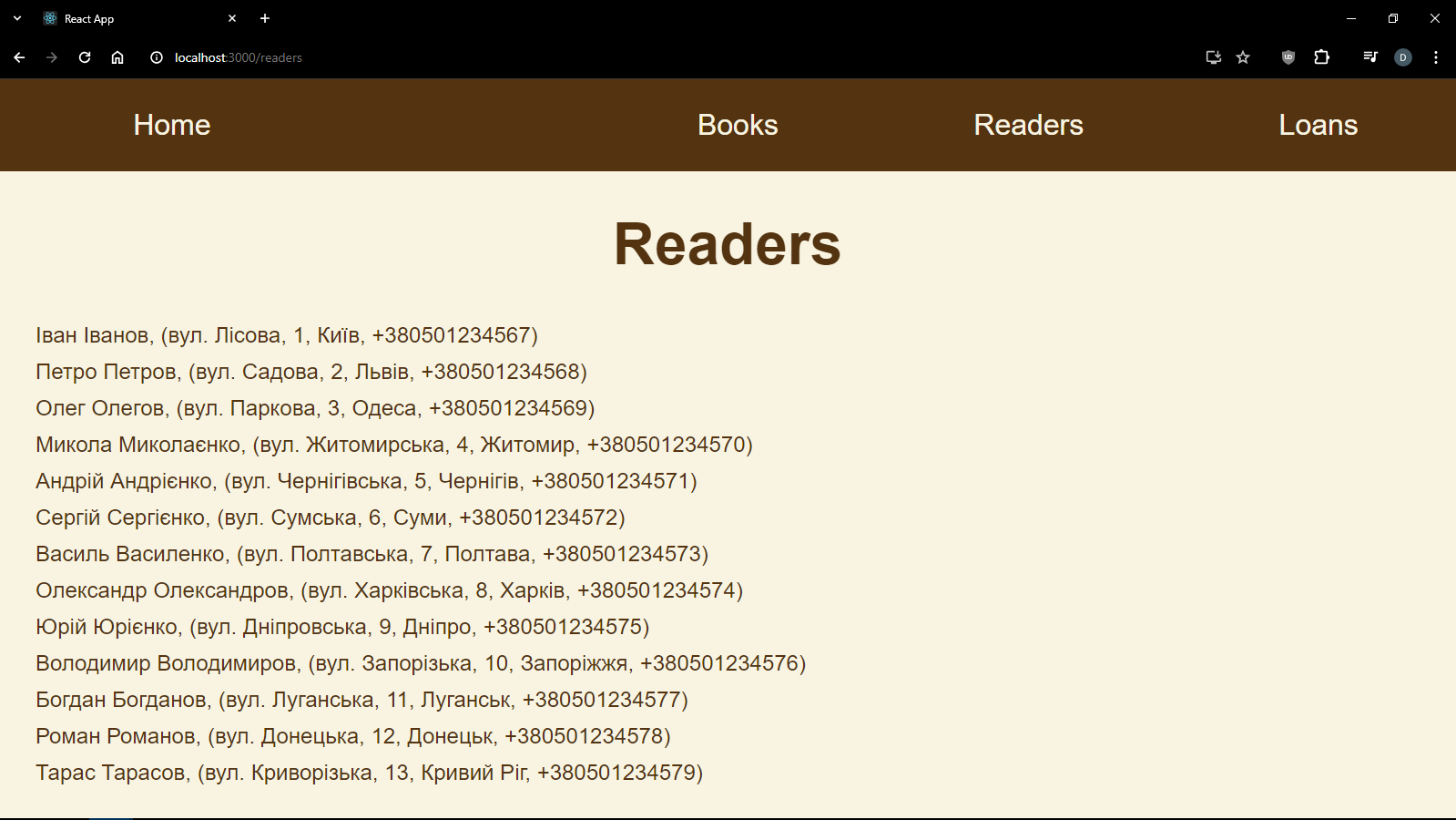
* Головна



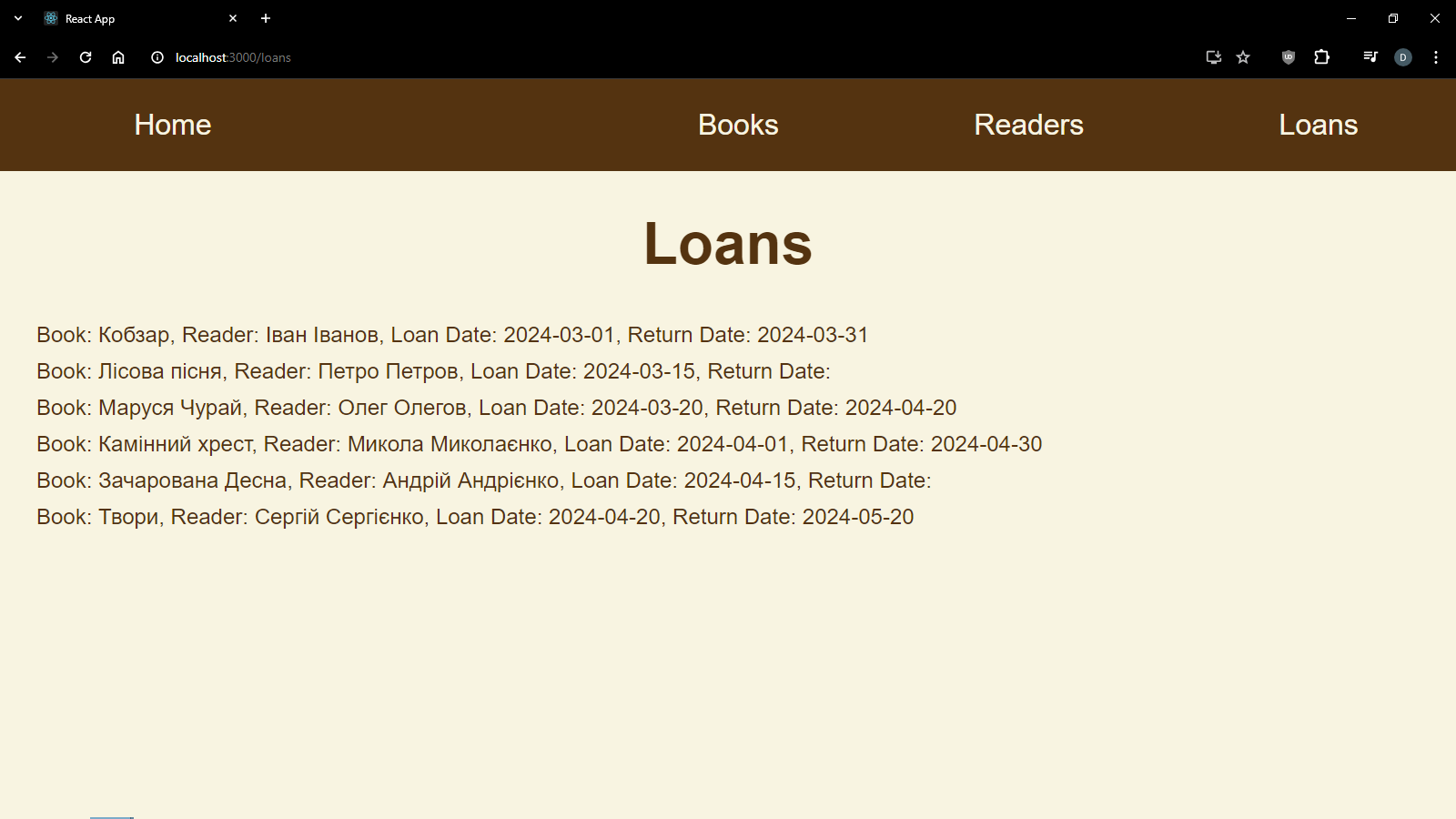
* Книги (виведення книг на сторінку відбувається з бази даних в MongoDB)



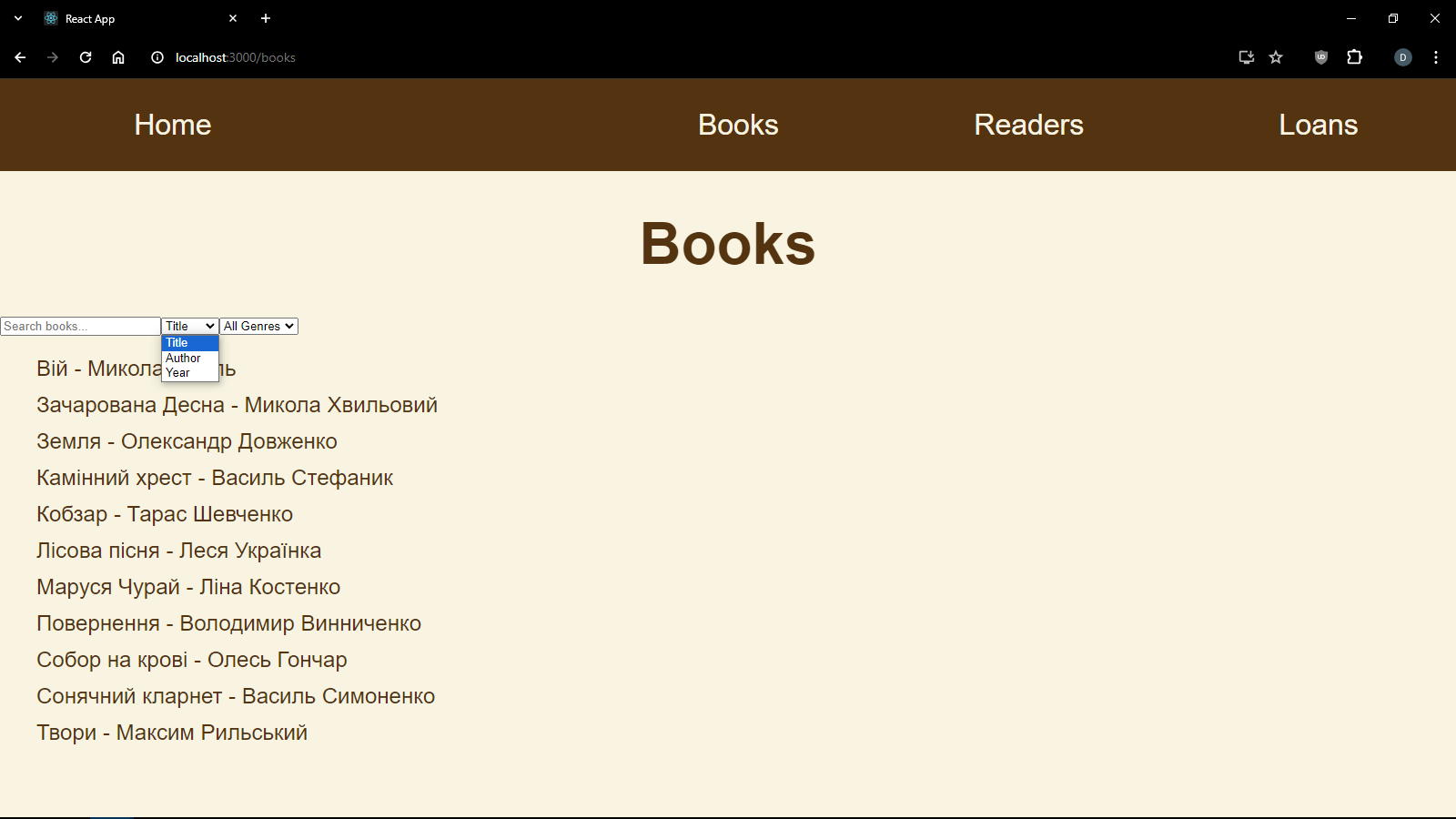
* Читачі

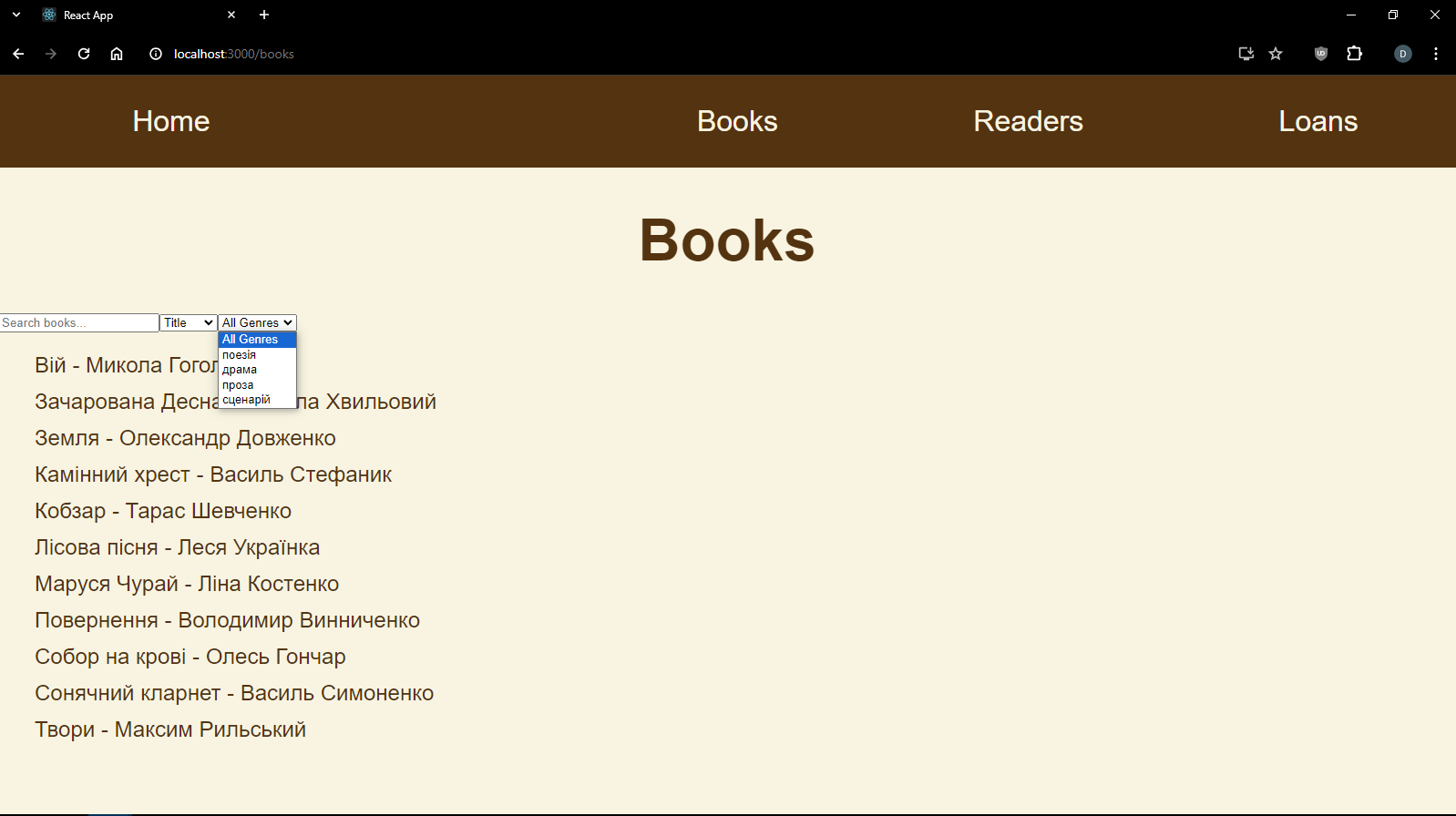


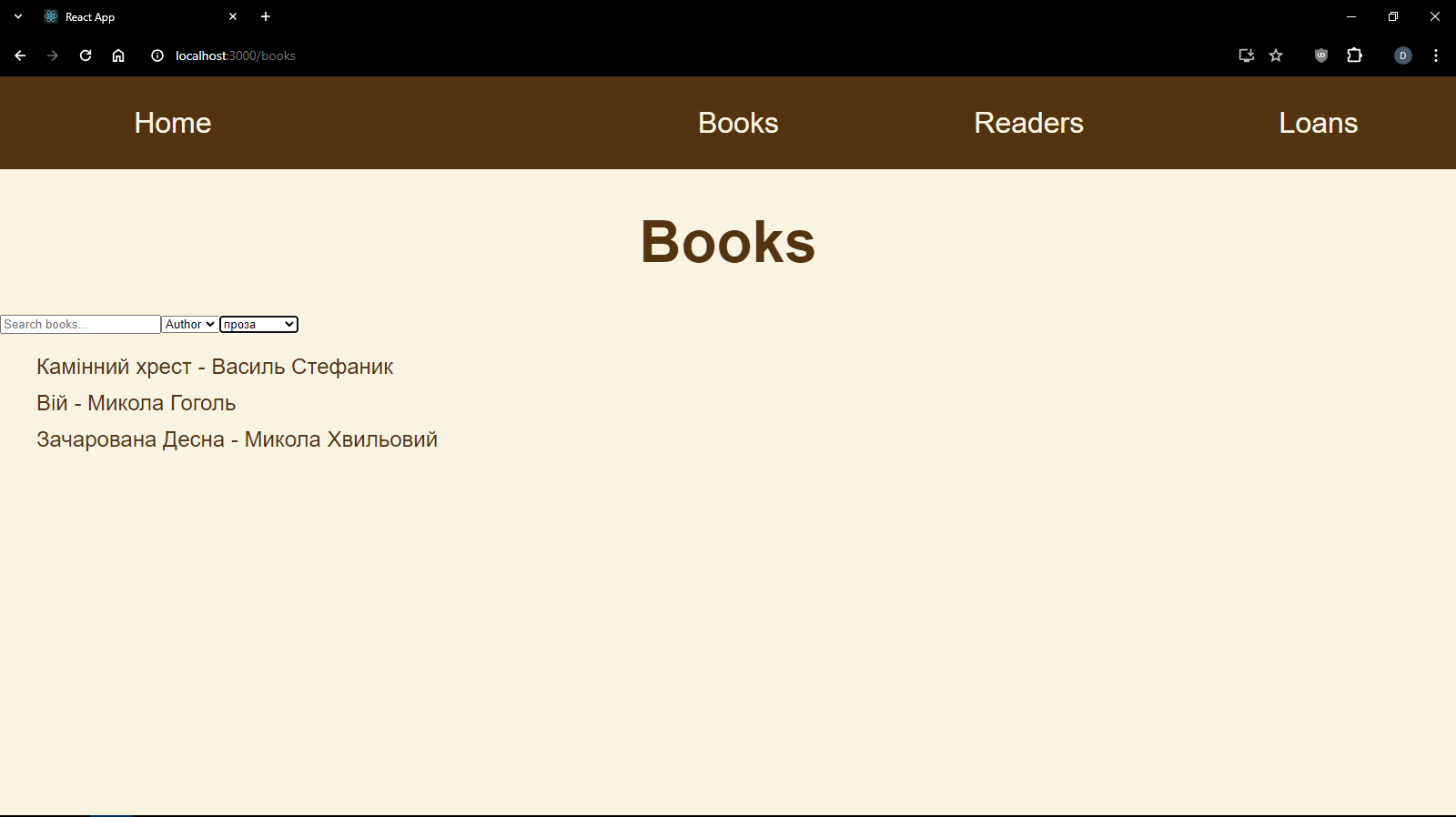
* Позики (книга, яку позичили та читач, що позичив, беруться по id з books та readers колекцій відповідно)

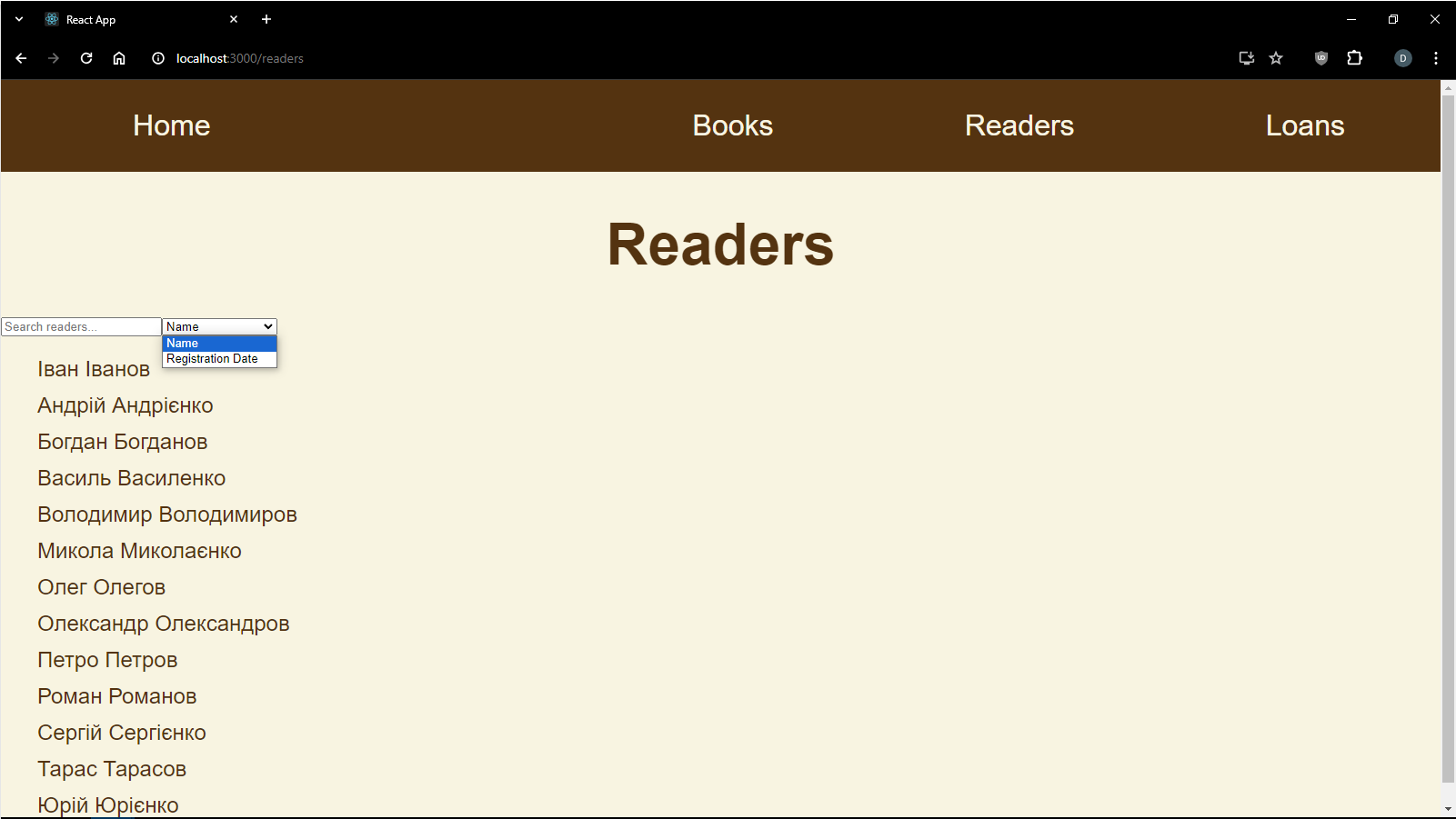


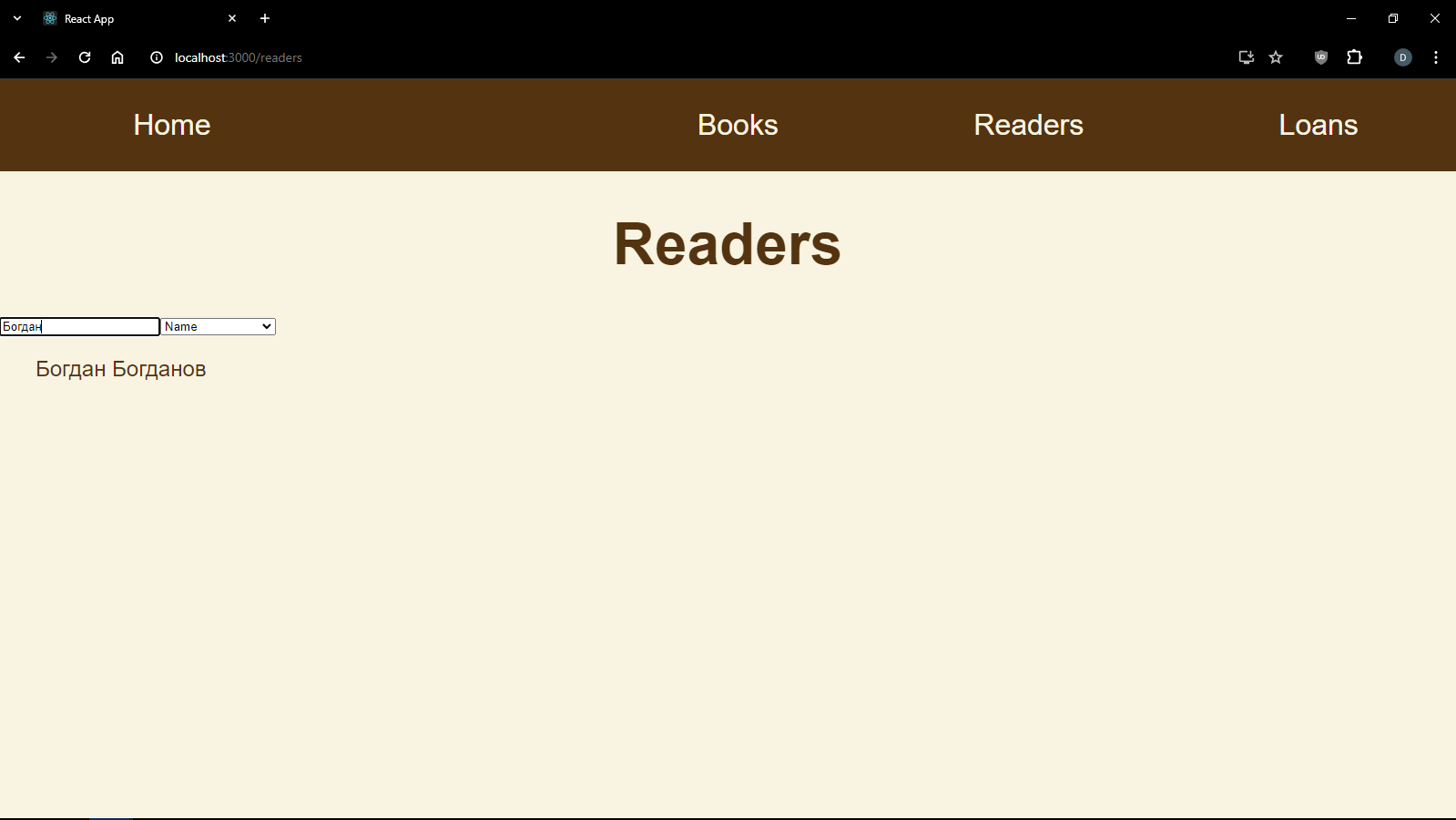
Додаємо функції пошуку, сортування та фільтрації даних. Книги можна сортувати за заголовком, автором та роком, фільтрувати за жанрами та шукати книгу за назвою або автором. Читачів можна шукати та фільтрувати за іменами та датами реєстрації.











**Репозиторій**: <https://github.com/DenysChornokon/db>

**Висновок:**

В результаті виконання даної лабораторної роботи було розроблено повнофункціональну інформаційну систему для обробки та візуалізації даних в області “Бібліотека”.